UNCLASSIFIED

AD NUMBER
ADB008128
LIMITATION CHANGES
TO: **Approved for public release; distribution is unlimited.**
FROM: **Distribution authorized to U.S. Gov't. agencies only; Test and Evaluation; 08 DEC 1975. Other requests shall be referred to Assistant Chief of Staff (USAF) for Studies and Analysis, Washington, DC.**
AUTHORITY
USAF ltr 31 Oct 1977

THIS PAGE IS UNCLASSIFIED

THIS REPORT HAS BEEN DELIMITED
AND CLEARED FOR PUBLIC RELEASE
UNDER DOD DIRECTIVE 5200.20 AND
NO RESTRICTIONS ARE IMPOSED UPON
ITS USE AND DISCLOSURE.

DISTRIBUTION STATEMENT A

APPROVED FOR PUBLIC RELEASE;
DISTRIBUTION UNLIMITED.

SCI 5997

2

AD B008128

METHODOLOGY TO EVALUATE STRATEGIC COMMAND AND CONTROL SYSTEMS

FINAL REPORT

Distribution limited to U.S. Gov't. agencies only; Test and Evaluation; 8 Dec 75. Other requests for this document must be referred to

PREPARED FOR:

ASSISTANT CHIEF OF STAFF, STUDIES AND ANALYSIS
HEADQUARTERS UNITED STATES AIR FORCE

AFOSR CONTRACT NO. F44620-73-C-0033

AD No.
DDC FILE COPY

DECEMBER 1973

DDC
DEC 17 1975
A

SYSTEMS CONTROL, INC.
260 SHERIDAN AVENUE
PALO ALTO, CALIFORNIA 94306

COPY 20 OF 25 COPIES

Letter on file

White Section
Buff Section

B

(14) SCI-5997

SYSTEMS CONTROL, INC.
260 SHERIDAN AVENUE
PALO ALTO, CALIFORNIA 94306

TELEPHONE (415)
327-9333

(12) 71p.

(6)

METHODOLOGY TO EVALUATE STRATEGIC COMMAND AND CONTROL SYSTEMS.

(9) FINAL REP[redacted]RT.,

Prepared for:

ASSISTANT CHIEF OF STAFF, STUDIES AND ANALYSIS
HEADQUARTERS UNITED STATES AIR FORCE
Wash DC 20330

AFOSR CONTRACT NO. F44620-73-C-0033

(15)

(11) Feb[redacted]74

Distribution limited to U.S. Gov't. agencies only; Test and Evaluation; DEC 1975. Other requests for this document must be referred to

DDC
RECEIVED
DEC 17 1975
A

Prepared by:

(10) Lewis Meier,
Ronald E. Foerster
Elizabeth Ducot

Approved by:

A. J. Tether

Anthony J. Tether, Director
Systems Technology Branch

R P Wishner

Richard P. Wishner
Vice President

Approved by:

David M. Salmon

David M. Salmon
Project Supervisor

389 333

TABLE OF CONTENTS

TABLE OF CONTENTS (CONT'D)

LIST OF FIGURES

LIST OF TABLES

I. INTRODUCTION

This document is the Final Report provided by Systems Control, Inc. (SCI) to the Office of Assistant Chief of Staff Studies and Analysis, United States Air Force, under contract F44620-73-C0033, "Development of Methodology To Evaluate Strategic Command and Control Systems." The project was technically directed by the Command, Control and Reconnaissance Division, Director of Strategic Offensive and Defensive Studies, Assistant Chief of Staff Studies and Analysis, HQ USAF, which provided valuable guidance and data throughout the conduct of this research.

The principal objective of the effort was to aid the Command, Control and Reconnaissance Division in the development of methodology for evaluating the survivability and effectiveness of Strategic Command and Control (C^2) Systems. While many issues must be considered when analyzing the capabilities of Strategic C^2 Systems, the principal one addressed in this study involves the communication systems which would be employed in the conduct of a nuclear war. The availability and time responsiveness of the communication systems, the vulnerability of the systems to nuclear and electronic warfare (EW) attack, and the relationship between communication system effectiveness and Strategic Command and Control effectiveness, are examples of some important issues that the methodology addresses.

The main element of the methodology is a Strategic C^2 effectiveness model that evaluates communication system effectiveness in terms of Strategic C^2 effectiveness while accounting for the salient factors which have a significant impact on system effectiveness. The model computes the statistics of message arrival as a function of time for a given network specification and a given enemy nuclear attack on the network. The model consists of two separate but interrelated computer programs and it is computationally efficient, allowing Strategic C^2 planners to conduct extensive parametric studies of various offense tactics and defense communication systems and operating procedures. Factors that impact system effectiveness and that are accounted for in the model include:

- Signal strength attenuation resulting from atmospheric nuclear detonations

- Noise from enemy jammers
- Survivability (i.e., probability of being operable) of links and nodes under nuclear attack
- Time delays incurred by messages passing through communication system nodes
- Transmission delays of messages transmitted on low data rate links and probability of correct message receipt on these links
- Directionality of communication links (i.e., one-way or two-way communication)
- Survivability of strategic forces at the time of message receipt

An important fact to note is that communication system effectiveness (as well as Strategic C^2 effectiveness) can only be estimated in a statistical sense due to the stochastic nature of the processes involved. The model therefore must be capable of generating expected values and probability distributions for all parameters of interest such as the number of nodes in a specified class that receive a message prior to a specified time, or the time of message reception at specified nodes within the network. The probability distributions, which take full account of correlated events that inevitably arise in message propagation through a communication network, are of particular importance to Strategic C^2 planners. These distributions are necessary to allow proper interpretation of the effectiveness data generated by the model; for example consider the following two probability distributions: (1) all 20 nodes of a class receive the message by a specified time with probability 0.5 or none of the 20 nodes of that class receive the message by that time with probability 0.5; and (2) all 20 nodes of the specified class receive the message by the specified time independently with probability 0.5. These two distributions yield identical expected numbers of nodes receiving the message --10-- yet would have significantly different implications on Strategic C^2 effectiveness.

It is also important to note that while a static connectivity analysis has historically been employed to evaluate communication system effectiveness, such an approach is often misleading and inappropriate for evaluating Strategic C^2 communication system effectiveness. The static connectivity approach determines, given node and link probabilities of being operable, the probability that an operable path exists between any two specified nodes at that time (ignoring any link directionalization or delays along the path). In a Strategic C^2 communication network, the important issue is when a message arrives at destination nodes given that it was initiated at a specified time by a specified node. The "dynamic message propagation" approach to Strategic C^2 effectiveness analysis described in this report explicitly addresses this issue and can be differentiated from the static connectivity approach as follows:

- Link directionality is fully accounted for in the dynamic approach whereas it is not in the static approach.

- Probability distributions of message arrival times at specified nodes are computed in the dynamic approach whereas the probability an operable path exists between two specified nodes at a specified time is computed in the static approach.

- The relationship between the position of a message in the network and the times of link and node destruction (e.g., by nuclear attack) is fully accounted for in the dynamic approach (i.e., destruction of a relay node <u>after</u> a message passes through it does not influence the statistics of propagation of that message) whereas it is not treated correctly in the static approach.

The first point above is important because many links in a Strategic C^2 communication network are one-way. The second point is important because, for a variety of reasons, propagation of messages through a Strategic C^2 communication network can take a significant amount of time (i.e., minutes instead of fractions of seconds) and it is end-to-end message propagation time that plays a critical role in determining strategic command and control effectiveness. The third point is important because during the time interval a Strategic C^2 communication system is most needed and messages are propagating through the network, it is also susceptible to attack and hence may suffer outages at links and nodes; hence, when generating a candidate offense attack on the network in order to properly test the network effectiveness, one must not only determine which links and nodes to attack but also when the attacks must occur to be effective in degrading the message arrival times at destination nodes.

Systems Control, Inc. has contributed to the development of the dynamic approach for evaluating Strategic C^2 effectiveness by assisting the Command Control and Reconnaissance Division in formulating an appropriate methodology and by defining and developing the two computer programs upon which the methodology is based--the Network Status Model and the Dynamic Network Simulator. The Network Status Model computes, as a function of time, the status of each node and link in the network. Nuclear effects calculations, electronic warfare calculations, and hence all calculations relating to network geometry and node/link vulnerability are performed by this model. The Dynamic Network Simulator, which requires as input the Network Status Model output, propagates messages through the overall network and computes the statistics relating to Strategic C^2 performance. Most of the effort in this initial study was focused on the development of the Dynamic Network Simulator; this work will be fully described in succeeding chapters. Development of the Network Status Model is on-going and hence will not be described in great detail in this report.

The next chapter more fully discusses the problems of modelling and analyzing the effectiveness of Strategic C^2 communication networks, and the methodology developed for addressing the problem. Succeeding chapters describe the methodology in more detail.

II. THE STRATEGIC C^2 PROBLEM AND SOLUTION METHODOLOGY

The Strategic C^2 networks of concern here link the National Command Authority (NCA) to the strategic forces--at present consisting of the Strategic TRIAD: Minuteman Missiles, SAC Bombers, and Polaris Submarines. The Strategic C^2 communication problem is concerned with the development and operation of communication systems to be employed in the conduct of a general nuclear war. The problem may be broken into three subproblems: the attack warning problem--determination that an enemy attack has been initiated; the delivery of attack orders problem--transmission of the Emergency Action Message (EAM) from the NCA to the strategic forces; and the battle management problem--the exchange of messages between the NCA and the Strategic forces during the course of the war. This report is primarily concerned with modelling and evaluation of communication systems employed in delivering attack orders.

Description of the Strategic C^2 Communication Network

A Strategic C^2 network can consist of a variety of land-based, airborne, seaborne and satellite nodes electromagnetically linked. Potentially, the network may have land lines, or electromagnetic links in any of the bands ELF-LF, MF-HF, and VHF-EHF. However, the low frequency links (ELF-LF) and the high frequency links (VHF-EHF) are most likely to survive a nuclear attack and hence only these links will be explicitly considered here. However, the methodology developed herein is also applicable to the other forms of links. Because the nodes and links of a Strategic C^2 network are subject to possible destruction and disruption by nuclear bursts and enemy jamming, a multiplicity of paths are generally provided between the NCA and the strategic forces--in particular, two nodes are often connected by two or more links of differing types.

Properties of Nodes and Links. The propagation of the EAM through a Strategic C^2 Network depends upon: (1) the survivability of nodes and their times of destruction, (2) the time delay at each node between receipt and retransmission of the message due to nodal processing procedures such as decoding and coding, and (3) the delay for each link, that is, the time from initial transmission of the message at the sending node until it is completely and correctly received at the receiving node. Link delays are a function of the length of the EAM, the form of link, and the probability of incorrect reception of the message. For example, the transmission time at ELF-LF may be minutes and a message may be transmitted several times before it is correctly received; hence the link delay may be a multiple of the message transmission time or even infinity if the link is broken. The message transmission times at VHF-EHF are much smaller but still not negligible.

Types of Nodes. A node may be moving or stationary and may be classified as being in one of three classes: source nodes, ordinary nodes, and special nodes. A source node is a node at which the EAM is initiated; typically there will be only one source node in a Strategic C^2 network. An ordinary node is neither the source, nor a destination, of the EAM; it serves to relay the message and it is always linked to other nodes from which it can receive the message and to which it can send the message. A special node is any node for which the user wants statistics on arrival of the EAM.

Types of Links. A link between two nodes may be one-way or two-way, and moving or stationary depending on whether or not one of the two nodes it connects is moving. For present purposes it is convenient to separate all links into four classes: ELF-LF links, VHF-EHF links, satellite links, and special links. ELF-LF and VHF-EHF links are stationary (aircraft and seaborne nodes are assumed stationary for time and distance scales of interest here), while satellite and special links are moving VHF-EHF links. The first three types of links are assumed to be operating prior to injection of the EAM into the network. Special links correspond to nodes which move a great distance during the simulation and do not become operable until a fixed time after the sending node of the link receives the message. Special links may be treated in a manner similar to fixed VHF-EHF links by creating fictitious links and nodes: the message travels instantaneously over a fictitious perfect link to a fictitious node which than becomes a sending node for the special link after a fixed delay corresponding to the time required for the link to become operable.

Decomposition of the Problem.

The status of the nodes (i.e., dead or alive) and the status of the links (i.e., their ability to transmit the EAM) at any given time is a function of certain external events (such as nuclear bursts and jamming) caused by complex physical phenomena, while the progress through the network of the EAM given node and link status is statistical. Therefore it is convenient and desirable to decompose the problem of evaluating Strategic C^2 effectiveness into two parts: determination of network status as a function of time followed by determination of network effectiveness in propagating the EAM to the strategic forces. This decomposition is possible because while network effectiveness depends upon the time history of network status, the opposite is not true. The separation between network status and network effectiveness calculations is somewhat arbitrary and should be based on convenience and--since the calculations will be made on a digital computer--on computer efficiency considerations.

Determination of Network Status. To determine the network status one must compute the status of each node and each link at each time increment of interest. (The assumption is made here that time may be quantized into fixed increments.) An important question is whether one should compute the status of all nodes and links at a given time and cycle through the times of interest or whether one should compute the status of a given node or link for all times and cycle through the nodes and links. Since the computations made to determine node or link status at a given time may provide information that is useful to determine the status of that node or link at a later time, the latter organization is preferable. Figure 2.1 is a high-level flow chart of a computer program for determining network status (the Network Status Model) utilizing this organization. Table 2.1 is a list of inputs required by this model, and Table 2.2 is a list of outputs from this model. Link and node status definitions and calculations will be more fully discussed in later sections.

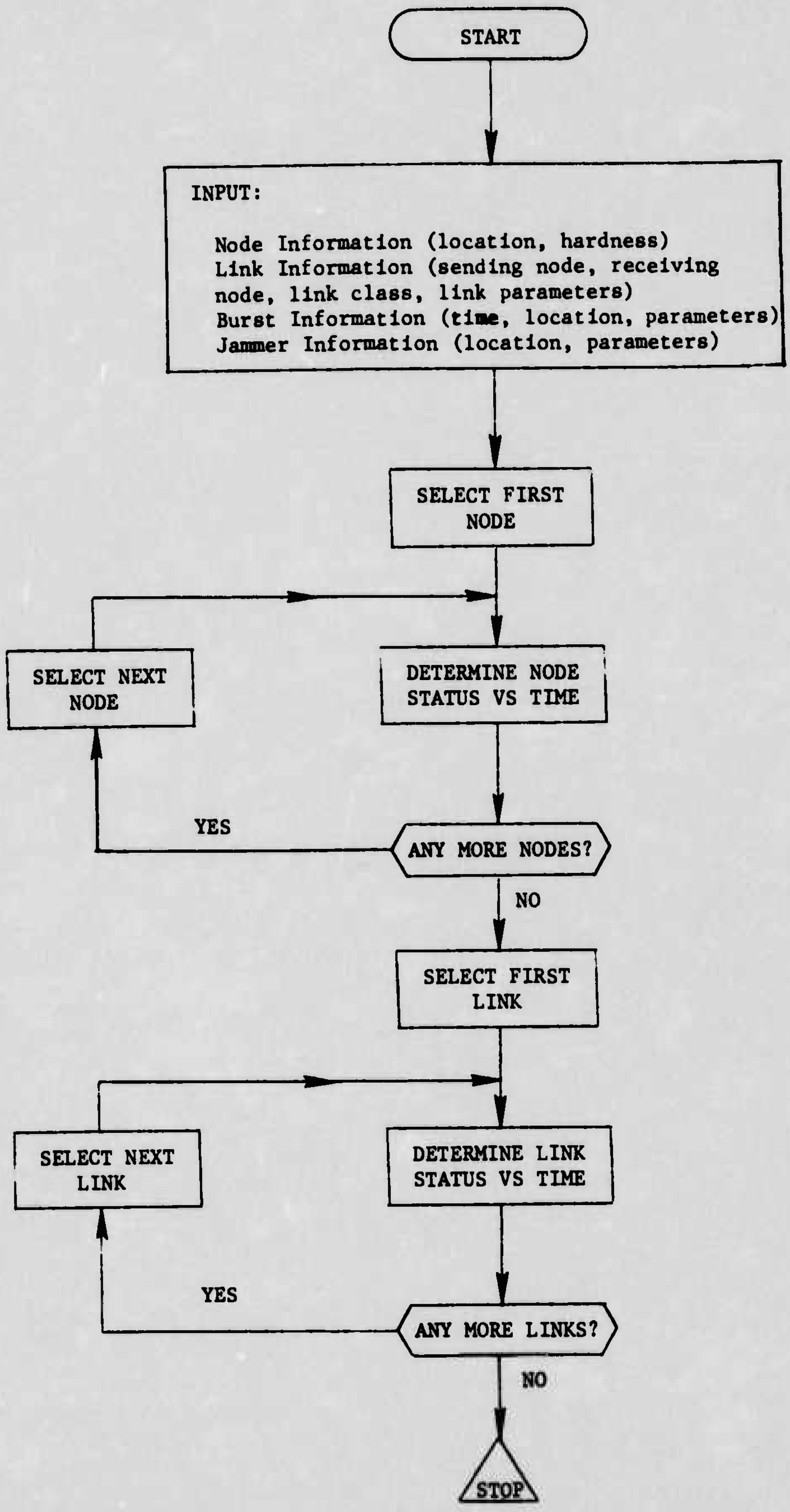


FIGURE 2.1 NETWORK STATUS MODEL

TABLE 2.1 INPUTS TO THE NETWORK STATUS MODEL

1. Node Information

   a. Location: latitude, longitude, altitude of each node.
   b. Hardness: V_N number for fixed nodes, (time, cumulative probability of destruction) pairs for other nodes.
   c. Node Delay: Minimum node delay and interval of node delays for all nodes.*
   d. Node Class for each Special Node.*
   e. Receipt Value vs. Time: (time, value of message) pairs for each special node.*

2. Link Information

   a. Network Geometry: sending node and receiving node for all links.
   b. Link Class: either ELF-LF, VHF-EHF, satellite or special for each link.
   c. Message Transmission time for all links.*
   d. Time to Start of Message Transmission for all special links.*
   e. Parameters: transmitter effective radiating power, midband frequency, transmitting and spread bandwidths, modem type, data rate, receiving antenna gain and temperature, receiver line loss, side lobe gain, main lobe beamwidth, number of characters in digital alphabet, and number of phases for each link.†

3. Burst Information

   a. Location: latitude, longitude, altitude of each burst.
   b. Time of each burst.
   c. Parameters: yield, fission fraction, and black body radiating temperature for each burst, CEP of the warhead.

---

* These inputs are not used by the Network Status Model but merely transferred to the outputs for use by the Dynamic Network Simulator.

† This is a preliminary list of link parameters. Because of the large number of parameters and links a small number of link types will be defined and parameters stored in the model for each type; input will merely specify the link type.

TABLE 2.1 Inputs to the Network Status Model (Cont.)

4. Jammer Information

   a. Location: latitude, longitude, altitude of each jammer.

   b. Parameters: effective radiated power, midband frequency, and bandwidth for each jammer.

TABLE 2.2 OUTPUTS OF THE NETWORK STATUS MODEL

(Inputs Required by the Dynamic Network Simulator)

1. Node Information

   a. Node Status vs. Time: (time, cumulative probability of destruction) pairs for each node.

   b. Node Delay: Minimum node delay and interval of node delay for all nodes.*

   c. Node Class for each special node.*

   d. Receipt Value vs. Time: (time, value of message) pairs for each special node.*

2. Link Information

   a. Network Geometry: sending node and receiving node for each link.*

   b. Link Class: either ELF-LF, VHF-EHF satellite, or special for each link.*

   c. Message Length for each ELF-LF link.*

   d. Time to start of Message Transmission for each special link.*

   e. Link Status vs. Time: (time, probability of acceptable message) pairs for each ELF-LF link, (time, probability of availability) pairs for each other link.

---

* These outputs of the Network Status Model are inputs to it and are unchanged by it.

Determination of Network Effectiveness. The propagation of an EAM through a Strategic C^2 network is stochastic because node destruction, node delays, and transmission delays are random processes. (A description of the models developed for these random processes is given in Section III.) Network effectiveness is determined from the statistics of message arrival at the special nodes. (A discussion of appropriate statistics is given in Section V.) Two general methods exist for determining the statistics of a stochastic process: statistical simulation and Monte Carlo simulation. In a statistical simulation the probability distribution of the state of the process is propagated forward in time; whereas in a Monte Carlo simulation a number of runs are made with the random processes simulated by appropriate use of random number generators. Unfortunately, for all but the simplest processes the number of states of the system--hence the dimension of the probability distribution--can be very large rendering stochastic simulation impractical.* Fortunately, use of the shortest path algorithm (as detailed in Section IV) permits each run of a Monte Carlo simulation to be made efficiently so that sufficient runs may be made to obtain reliable statistics with a reasonable amount of computer time (the number of runs required is discussed in Section V). Figure 2.2 is a high level flow chart of a computer program (the Dynamic Network Simulator) for determining network effectiveness utilizing Monte Carlo simulation. Table 2.2 is a list of inputs required by this model, and Table 2.3 is a list of outputs generated by this model. For each node a determination of node destruction time (possibly infinity) is made prior to message propagation. On the other hand, to avoid computing node and link delays at every time increment, when in fact they are needed only at the time the message arrives at the node or link, these delays are computed within the message propagation algorithm.

Efficient Use of Network Status Model and Dynamic Network Simulator. Because the Network Status Model contains nuclear effects codes and generally requires

---

* For each node of a strategic C^2 network, at a given time it must be known if and when a message has arrived. Thus, at time k each node can be in any one of k + 1 states and a system of m nodes can be in any one of $(k + 1)^m$ states--an astronomical number for typical values of k and m.

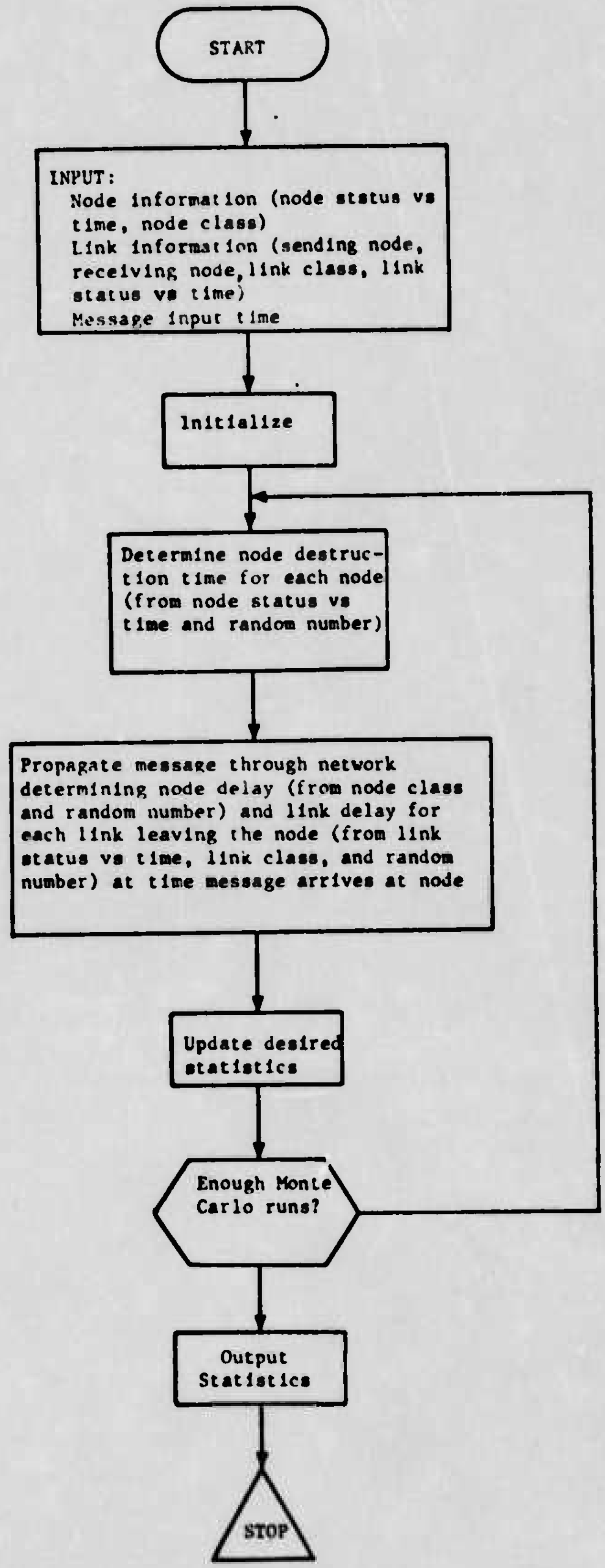


FIGURE 2.2 DYNAMIC NETWORK SIMULATOR

TABLE 2.3 OUTPUTS OF THE DYNAMIC NETWORK SIMULATOR*

1. Time Period Histograms (See Section V).
2. Arrival Time Distributions (See Section V).
3. Probability of any Event that can be defined as occurring or not occurring on a Monte Carlo Run.
4. Probability Distribution Over a Set of Mutually Exclusive Events (Outputs 1, 2, and 3 are special cases of output 4).
5. Expected Value of Performance Index (See Section V).
6. Expected Value of any Random Variable whose value may be defined for each Monte Carlo Run (Outputs 3 and 5 are special cases of output 6).
7. For any Monte Carlo run a map of the dominant Nodes and Links that actually carry the message.
8. For any link, the probability it will carry the message (a special case of output 3).
9. For any node, the probability it will relay the message (a special case of output 3).

---

* This list is not meant to exhaust the possibilities of outputs, but rather to suggest some of them. Any output summarizing the results of all Monte Carlo runs may be viewed as a set of outputs of Type 6.

a significant amount of computer time, while the Dynamic Network Simulator is computationally much faster, it is important to utilize the two models in an efficient manner. Typically, one will be interested in a baseline Strategic C^2 network and several proposed modifications of the baseline. To perform trade-off studies efficiently it is desirable to develop an augmented baseline network such that the baseline and each of the modified baselines is a subnetwork of the augmented baseline network. The Network Status Model is then used to obtain the status of the augmented baseline network and a series of simulations are made with the Dynamic Network Simulator in which, for each simulation, only those nodes and links comprising the system of current interest are considered. Furthermore, it may be desirable to perform a large number of trade-off studies with the Network Status Model embodying greatly simplified nuclear codes followed by selective use of detailed nuclear codes to verify key results obtained with the simplified codes.

III. DETERMINATION OF NODE AND LINK PROPERTIES

In the previous section it was pointed out that propagation of the EAM through the Strategic C^2 network is dependent upon how long each node survives, the message delay at each node, and the transmission delays along each link. This section describes how these quantities are determined.

Node Properties

Two node properties are of interest: node survival and node delay. These properties are individually discussed below.

Node Survival. Consider a given node. For each nuclear burst in the vicinity of the node there is a non-zero probability the node will be made inoperable. This probability depends upon the size and location of the burst and the hardness and location of the node. A model for determining this probability (for over-pressure kill only) is given in Appendix A. By use of this model a number of (time, probability of destruction) pairs (t_i, q_i) may be generated with each time t_i corresponding to each burst that has a non-negligible probability q_i of destroying the node at time t_i, $t_1 < t_2 < \cdots$. From the pairs (t_i, q_i) a set of time-cumulative probability of destruction pairs (t_i, q_i^*) can be generated--where q_i^* is the probability that the node is destroyed by the burst at time t_i or by any other burst prior to time t_i. The q_i and q_i^* obey the following relationships:

$$q_1^* = q_1,$$

$$q_{i+1}^* = q_i^* + (1-q_i^*)\, q_{i+1} \text{ for } i \geq 1 \tag{3-1}$$

To determine the time of destruction of a node a random unit (i.e., random number uniform on [0,1]) is generated and compared with the q_i^* s. The time of destruction is the earliest t_i for which q_i^* is greater than the random number.

Computation of q_i, the probability of destruction by a single burst, is performed in the Network Status Model. The calculations represented by (3-1) can be made logically in either the Network Status Model or in the Dynamic Network Simulator--a somewhat arbitrary decision was made to perform these calculations in the Network Status Model. In the present version of the Network Status Model (t_i, q_i^*) pairs are computed only for ground-based nodes--the user must specify (t_i, q_i^*) pairs for other node types* (e.g., satellite and aircraft nodes). During each Monte Carlo simulation of the C^2 system the time of node destruction is determined by the Dynamic Network Simulator prior to use of the shortest path algorithm.

Node Delay

Node delays are currently input for each node. They are assumed to be uniformly random between minimum and maximum values $\Delta t_{Minimum}$ and $\Delta t_{Maximum}$ $(= \Delta t_{Minimum} + \Delta t_{Interval})$. The actual node delay Δt is determined for each Monte Carlo run in the Dynamic Network Simulator by selecting a random unit r for each node prior to the time it is to relay the EAM. At the time it is to relay the message the delay is determined from

$$\Delta t = \Delta t_{Minimum} + r(\Delta t_{Interval}) \tag{3-2}$$

For each fictitious node generated by a special link, the node delay is equal to the time after message receipt that the link becomes active (see Section II).

Link Properties

It is assumed that the received signal strength and noise can be modelled by a Gaussian distribution. Accordingly, the mean and standard deviation of signal and noise at the receiving nodes must be computed. For ELF-LF and VHF-EHF links these values are used in different manners to compute link delays.

Signal and Noise. The signal strength at the receiving node of a link in the absence of nuclear effects is a function of sending and receiving node

---

* There are standard procedures for doing this.

locations and the link class (which determines transmitter and receiver characteristics, frequency, etc.). Nuclear bursts create absorbing regions (the only effect thus far considered) that lower the signal strength at the receiver. The amount of signal attenuation is a function of burst size and location in addition to the link location and class. Noise at the receiver is the sum of ambient (atmospheric) noise (a function of link class) and noise from jammers (a function of jammer location and type, and link location and class). Description of the models required to determine nuclear-free signal strength, nuclear attenuation levels, and ambient and jamming noise levels is beyond the scope of this report. Suffice it to say that the requisite computations are to be made in the Network Status Model.

ELF-LF Links. Given a particular value of the Signal to Noise ratio (S/N) there is a resulting Probability of Character Error (PCE) which is a characteristic of the type of communication link. Given a particular value of PCE there is a resulting probability that the message will be accepted (PAM) which is a function of how the message is structured into characters and the error correction procedures that are employed. Thus there is a one to one relation between S/N and PAM which is defined by series application of the two curves relating PCE to S/N and PAM to PCE. If the S/N varies over the message transmission interval, it is appropriate to use the smallest S/N value over the interval since even a short drop in S/N may result in many character errors which render the overall message unacceptable. Thus starting from the Gaussian distribution for S/N it is straightforward to obtain a curve of PAM versus time, which yields the probability that a single transmission of the message starting at a given time will be acceptable when it is completely received.

Consider a given ELF-LF link and assume that a single transmission of the EAM takes Δt time units, that the EAM is first transmitted at time t, and that it is sent repeatedly until it is correctly received at the receiving node or the final time of interest, t_f, has passed. Clearly, the message may be correctly received at $t + \Delta t$, $t + 2\Delta t, \ldots, t + (n-1)\Delta t$, $t + n\Delta t = t_f$: therefore the possible link delays are Δt, $2\Delta t, \ldots, n\Delta t$, where a delay of $(n+1)\Delta t$ implies that the message never reaches the receiving node (during the time interval of interest). To determine the link delay which actually applies on a given Monte Carlo run, the cumulative probability distribution $p_1^*, \ldots, p_n^*$ over the delay values can be determined from the curve of PAM versus time (using an equation similar to 3-1) and compared with a random unit.

Determination of the cumulative probabilities, p_i^*, may be made more complex if certain error correcting procedures are employed. For example, in one procedure the message could be correctly received at a given time on the basis of the last transmission, the last three transmissions, or the last five transmissions. The ability to correctly receive a message is then defined for each link class to be a set of three curves, one for each of 1, 3, and 5 transmissions, relating the probability of correctly receiving the message to PCE. Since PCE varies with S/N which in turn varies with time, the minimum value of S/N over the interval of interest should be used. For example: the probability $p_k(1)$ that the message was correctly received at time $t + k\Delta t$ on the basis of the k^{th} transmission alone is found by determining PCE using

$$S/N = \min_{i \text{ such that } t + (k-1)\Delta t \le t_i \le t + k\Delta t} S/N_i \tag{3-3}$$

and setting $p_k(1)$ equal to the probability of correct message receipt corresponding to this PCE using the single transmission curve. The probabilities $p_k(3)$ and $p_k(5)$ that the EAM was correctly received at time $t + k\Delta t$ on the basis of transmissions k-2, k-1, and k or k-4, k-3, k-2, k-1, and k, respectively, are determined similarly except that the minimizations of (3-3) should be carried out over the intervals $t + (k-3)\Delta t$ or $t + (k-5)\Delta t$ to $t + k\Delta t$ and the appropriate curves used. An equation similar to (3-1) can then be used to determine the cumulative probabilities p_k^* from $p_k(1)$, $p_k(3)$, and $p_k(5)$. Figure 3-1 is a flow chart of the complete procedure used to determine the p_k^*'s from the (time, S/N_i pairs.

If the data transferred between the Network Status Model and the Dynamic Network Simulator were the time-varying S/N, the logic of Figure 3-1 could be carried out as part of the shortest path algorithm as follows. A random unit could be selected for each link prior to determination of the p_k^*'s at the time

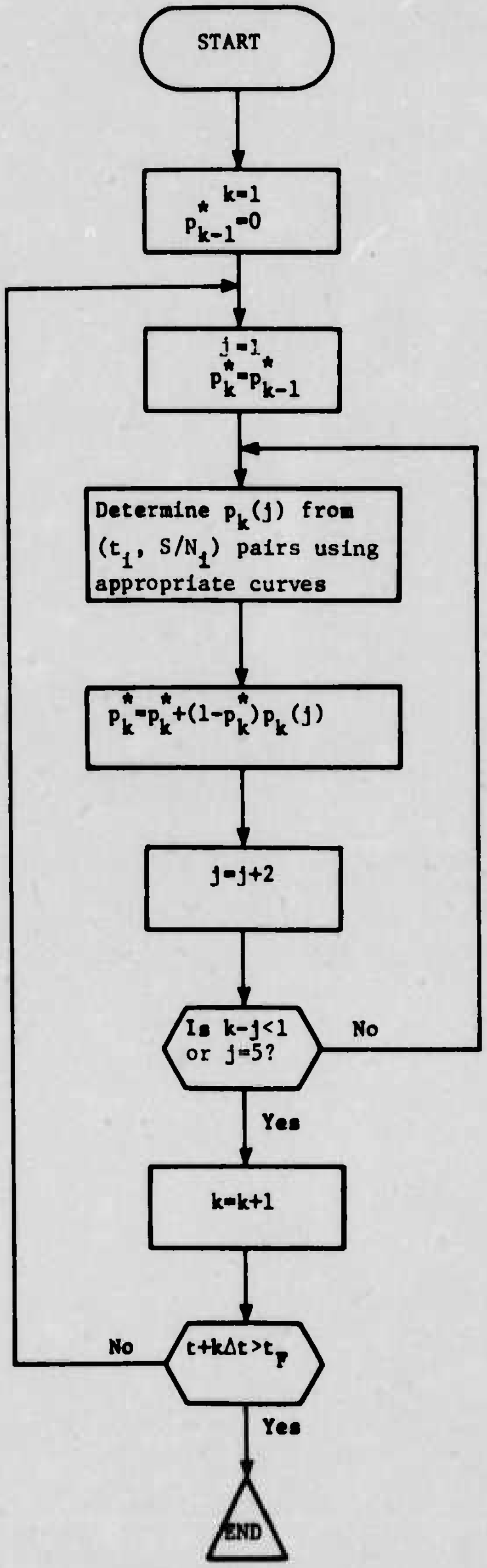


FIGURE 3.1 FLOW CHART OF PROCEDURE FOR DETERMINING THE p^*_k's

at which the message is to be sent over the link. Each time a new p_k^* is computed it would be compared with the random unit and when p_k^* exceeded the random unit, the message would be declared to be correctly transmitted with a link delay $k\Delta t$. A modified procedure is needed if the data transfer between the NSM and the DNS is the time-varying values of probability of acceptable message (PAM).

<u>VHF-EHF Links</u>. Because the EAM requires only a short transmission time at VHF-EHF, the links are modelled in a simpler fashion than the ELF-LF links From the signal and noise strength time histories, a time history of the probability P_t of successful transmission is determined in the Network Status Model in the form of (time, P_t) pairs (t_i, P_{t_i}). (Again, the means of performing this determination is beyond the scope of this report.) These pairs are utilized in the Dynamic Network Simulator as follows: For each VHF-EHF link a random unit is generated prior to the time the EAM is to be sent over the link. At the time t_i that the message is sent over the link, the random unit is compared with P_{t_i} corresponding to the largest t_i less than t. The message is successfully received depending on whether or not the random unit exceeds P_{t_i}.

IV. SIMULATION OF MESSAGE PROPAGATION

Given the statistics describing the node destruction times, the node delays, and the link delays, the propagation of the EAM throughout the Strategic C^2 Network must next be modelled. Section II indicated that this is accomplished through use of the shortest path algorithm (SPA). A description of this procedure is the major subject of this section, but first the motivation for using the SPA is discussed briefly.

Possible Methods of Analysis

Connectivity Algorithms. The classical tools for analyzing networks subject to node and link failures are connectivity algorithms which proceed as follows: based on probabilities of failure and random units, each node and link is determined to be either operable or inoperable. A connectivity algorithm then efficiently partitions the network into subsets of connected nodes such that communication between node A and node B is possible if A and B are in the same subset. By performing a Monte Carlo sequence of runs, statistics such as the probability that A can communicate with B, or other statistics relating to the connectivity of nodes in the network, can be determined. Reference 1 gives a detailed discussion of this approach as it applies to analysis of Strategic C^2 Networks. Unfortunately, connectivity algorithms suffer two shortcomings when applied to the problem of delivering attack orders. First, they only apply to two-way links--if A is connected to B then B is connected to A; secondly, they ignore dynamics--the times at which links or nodes become inoperable are not taken into account and the node and link delays are limited to the values of zero and infinity. Connectivity algorithms are more appropriate for steady state analyses of a network in which many messages are being sent and in which elements become operable and inoperable at random times. Attempts to extend the connectivity ideas to treat the delivery of attack orders problem proved fruitless because of these difficulties.

Shortest Path Algorithm. In studying the Strategic C^2 mission of delivering attack orders the basic assumption is made that each node, after an appropriate delay following message receipt, transmits the EAM along all links leaving it.

With this procedure it is clear that the message will arrive at every node in the network at the earliest possible time. The shortest path algorithm (SPA), known to network theorists (see Refenence 2), can be applied to solve for the resulting message arrival times. The path length of each link is defined to be the sum of the link transmission delay and the delay at the sending node. Since the SPA is a very efficient algorithm, it is quite suitable for modelling EAM propagation in a Strategic C^2 network. It should also be noted that while the SPA can be used to model dynamic message propagation it can also be used to determine static connectivity. This is done simply by setting delays either to zero or to infinity; see Appendix B for details.

Use of the Shortest Path Algorithm

A detailed description of the SPA as applied to the delivery of attack orders problem will be presented, followed by an example of its use. For the original derivation of the SPA the reader is referred to Reference 2.

Description of the SPA. The SPA is a procedure for labeling each node with the length of the shortest path between it and the source node. (If path length is time, the label for each node is the time it receives the EAM.) Suppose some of the nodes have been so labeled and refer to these nodes as "considered". Further suppose that each of the "unconsidered" nodes is labeled with the earliest time that it can receive the message over a single link from a "considered" node. Now take the "unconsidered" node whose label is smallest. Clearly it cannot receive the message any sooner than its label from any of the other "unconsidered" nodes; therefore it is appropriately labeled and may be removed from further consideration except that the label of each remaining "unconsidered" node must be updated by examining each link leaving it. This procedure of picking the "unconsidered" node whose label is smallest is initiated with all nodes "unconsidered", the source node labeled with the EAM injection time, and the remaining nodes labeled infinity. The procedure is repeated until all nodes have been "considered".

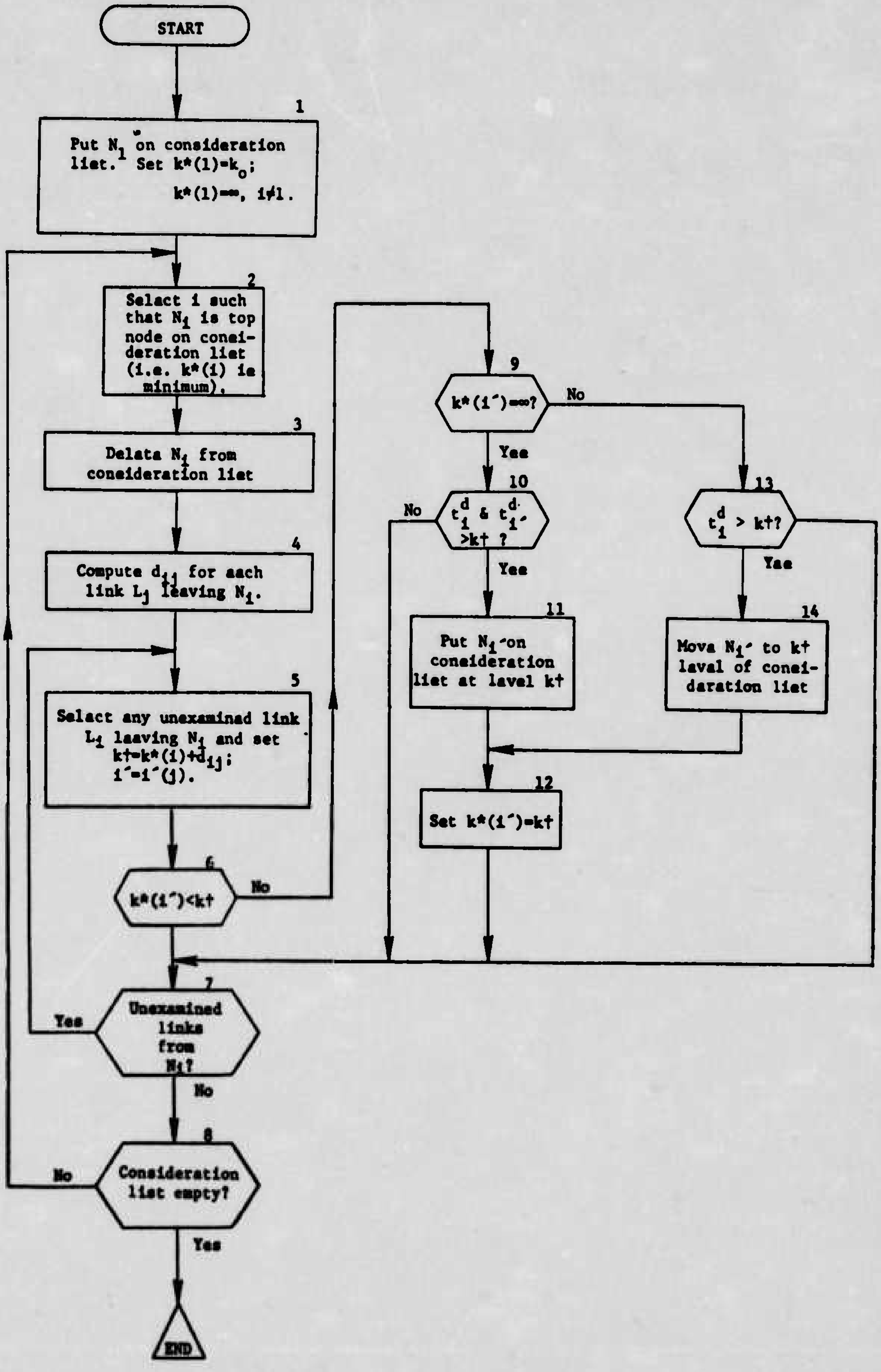


FIG. 4.1 FLOW CHART OF SHORTEST PATH ALGORITHM

Figure 4.1 is a flow chart of the SPA. The source node is N_1, N_i is the i^{th} node, t_i^d is the time of destruction for N_i, L_j is the j^{th} link, d_{ij} is the node delay for N_i plus the link delay for L_j (computed in the SPA as described in Section III), and $N_{i'(j)}$ is the node receiving the EAM from N_i over L_j. For convenience two lists are maintained by the program: a consideration list, in which all "unconsidered" nodes with finite label are listed in order of increasing label, and a node list in which the label of each node is listed. Note that in Box 10 a check is made to insure that both sending and receiving nodes will be operable until message transmission is complete. (Note further that, if $k*(i')$ is finite, only the sending node need be examined since $t_{i'}^d > k*(i')$ and only $k < k*(i)$ are of interest.)

A Simple Example. To illustrate the use of the SPA an example with two Monte Carlo runs will be presented. The network under consideration in this example is given in Fig. 4.2. The goal is to transmit a message from N_1 to N_3. Nominally L_1 and L_2 are fast links with a delay of 1 and L_3 a slow link with a delay of 3. The message may not be received correctly at the first transmission from a node; therefore on each link from the node the message is sent over and over until it is correctly received by the receiving node. Given the probability of correct reception for each transmission and the nominal delay for each link, the distribution of the delays along the link is readily determined. For this example these delays are taken to be independent. At time 3 a burst occurs that kills node 2 with probability $\frac{1}{2}$.

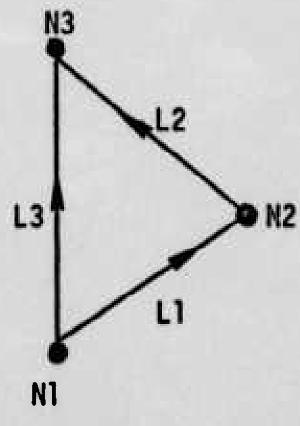


FIG. 4.2 EXAMPLE NETWORK

For Monte Carlo Run 1, a coin was flipped to see if N_2 was destroyed at time 3 and it was found that the node was not destroyed. The iterations taken by the shortest path algorithm are given in Table 4.1. During iteration 0, the algorithm was initialized by setting $k^*(1)=1$, $k^*(3) = \infty$ and placing N_1 on the consideration list at level 1. At the start of iteration 1, the top node on the consideration list was N_1. This node was deleted and the delays along L_1 and L_3 were computed using the appropriate distributions as given in Table 4.2. Two random numbers were selected; both were less than 0.667--therefore, d_{12} was set equal to 1, d_{13} was set equal to 3, N_2 was placed on the consideration list at level 2, and N_3 at level 4.

At the beginning of iteration 2, N_2 was at the top of the consideration list; it was deleted and d_{23} selected via Table 4.3. A random number was selected that lay between 0.889 and 0.907; therefore d_{23} was set equal to 3. Since

$$k^*(2) + d_{23} = 2 + 3 > 4 = k^*(3),$$

the position of N_3 on the consideration list and the value of $k^*(3)$ were left unchanged. At the start of iteration 3, only N_3 is on the consideration list. Because it has no output links, it is removed from this list and operation of the algorithm is complete. For this run of the algorithm, the earliest arrival times of the message at N_1, N_2, and N_3 were 1, 2, and 4, respectively.

For Monte Carlo run 2, a coin was flipped to see if N_2 was destroyed at time 3, and as before the node was not destroyed. The iterations taken by the shortest path algorithm are given in Table 4.4--iterations 0 and 1 were identical to run 1. During iteration 2, N_2 was considered and d_{23} was selected in accordance with the distribution given in Table 4.3. A random number was generated and found to lie below 0.667; hence d_{23} (2) was set equal to 1. Since

$$k^*(2) + d_{23} = 2 + 1 < 4 = k^*(3)$$

TABLE 4.1

Monte Carlo Run 1 of the Example

ITERATION		0	1	2	3
CONSIDERATION LIST	LEVEL 1	N_1			
	2		N_2		
	3				
	4		N_3	N_3	
ARRIVAL LIST (k*)	NODE 1	1	1	1	1
	2	∞	2	2	2
	3	∞	4	4	4

TABLE 4.2

Distributions of d_{11} and d_{13}

(First Subscript is Sending Node, Second Subscript is the Link)

j	ℓ	$\text{Prob}(d_{1j}=\ell)$	$\sum_{m=1}^{\ell} \text{Prob}(d_{1j}=m)$
1	1	0.667	0.667
	2	0.222	0.889
	3	0.074	0.963
	4	0.006	0.969
	5	0.005	0.974
	.	.	.
	.	.	.
	.	.	.
3	3	0.667	0.667
	6	0.167	0.834
	9	0.083	0.917
	.	.	.
	.	.	.
	.	.	.

TABLE 4.3

Distributions of d_{23}

ℓ	$\text{Prob}(d_{23}=\ell)$	$\sum_{m=1}^{\ell} \text{Prob}(d_{23}=m)$
1	0.667	0.667
2	0.222	0.889
3	0.018	0.907
4	0.016	0.923
5	0.0613	0.939
.	.	.
.	.	.
.	.	.

TABLE 4.4

Monte Carlo Run 2 of the Example

ITERATION		0	1	2	3
CONSIDERATION LIST	LEVEL 1	N_1			
	2		N_2		
	3			N_3	
	4		N_3		
ARRIVAL LIST (k^*)	NODE 1	1	1	1	1
	2	∞	2	2	2
	3	∞	4	3	3

N_3 was moved from level 4 to level 3 on the consideration list and k*(3) was set equal to 3. At iteration 3 only N_3 was on the consideration list. Since it has no links leaving it, it was deleted and this run was over--the earliest times of arrival of the message at N_1, N_2 and N_3 were 1, 2, and 3, respectively.

In this simple example no more than one node appeared in each level of the consideration list. For more complicated examples each level may contain many nodes at any given iteration of the algorithm. The order of nodes in a given level is immaterial, but placing the node at the proper level is required for proper operation of the algorithm.

V. DETERMINATION OF STRATEGIC C^2 SYSTEM EFFECTIVENESS

The previous three sections discussed Monte Carlo simulation of the Strategic C^2 Network mission of delivering attack orders. The present section addresses three questions: What are useful measures of system effectiveness? How many Monte Carlo runs are required to accurately determine these measures of system effectiveness? How can the number of Monte Carlo runs required be reduced?

Measures of System Effectiveness

It is convenient to divide the special nodes of the network into a number of classes; for example, all Minuteman nodes might be one class, all Submarine nodes a second class, and all Bomber nodes a third class. It is also appropriate to assign to each node in a class a time varying value for receipt of the EAM; for example, the Minuteman nodes might suffer attrition with time so that the value of receiving the EAM diminishes with time--in this case a suitable value for receipt of the EAM is the number of Minuteman Missiles remaining at the time of message receipt. Given the classes and their values for receipt of the EAM at various times, the total value of EAM receipt derived by members of a given class at a given time is readily determined for each Monte Carlo run. With these results for a number of Monte Carlo runs, several interesting measures of system performance may be determined; in particular, time period histograms of value and arrival time distributions are of interest.

Time Period Histograms. For this measure a particular time and set of classes are selected and a histogram of the total value of EAM receipt by nodes by the specified time is generated (i.e., probability of obtaining various values where value is appropriately quantized). For example, suppose the class of all Minuteman nodes is selected and the value of receiving the EAM is one at all nodes; then the histogram would correspond to the number of Minuteman nodes receiving the EAM by a particular time. In general, several families of histograms--each family for a given set of classes, each member of the family for a given time--may be of interest. Some examples of time period histograms are presented in Section VI.

Arrival Time Distributions. Consider a set of classes C_1, C_2, ..., C_n; assume a set of minimum values V_{MIN}^1; V_{MIN}^2, ..., V_{MIN}^n; and let $v_1(t)$, $v_2(t)$, ..., $v_n(t)$ be the values for EAM receipt by time t for each of the classes. The arrival time distribution is the cumulative distribution of the probability that $v_1(\tau) > V_{MIN}^1$, $v_2(\tau) > V_{MIN}^2$, ..., $v_n(\tau) > V_{MIN}^n$ at or before time τ For example let C_1 be the Minuteman nodes, C_2 be the Submarine nodes, and C_3 be the bomber nodes and suppose it is desired to know how soon n_1 Minuteman nodes, n_2 Submarine nodes, and n_3 bomber nodes receive the message. In this case, the value of receiving the EAM at each special node is set at unity and $V_{MIN}^1 = n_1$, $V_{MIN}^2 = n_2$, and $V_{MIN}^3 = n_3$. The result will be the cumulative distribution of when n_1 Minuteman nodes, n_2 Submarine nodes, and n_3 bomber nodes receive the message. Some example arrival time distributions are presented in Section VI.

Performance Indices. A single measure of network performance is the expected total value of special nodes which receive the EAM prior to specified times of interest. For example, suppose the classes are the three TRIAD forces as above and the value of EAM receipt is proportional to the size of the surviving force at the time of message receipt as weighted by the relative value of each type of force. The resulting performance index is the expected total value of the forces receiving the EAM. Again an example of a performance index is given in Section VI.

Statistical Analysis By Monte Carlo Simulation

It is clear that in attempting to compute a performance index one is attempting to compute the expected value of a random variable. The same is true when one is attempting to compute a given point on a time period histogram or an arrival time distribution. For example, consider the problem of computing the probability of deriving a given quantized EAM receipt value V on a time period histogram. Consider the random variable that has value one if the quantized EAM receipt value is V and zero otherwise. Then the probability of V is the expectation of this random variable. In general, two properties of a Monte Carlo simulation are of interest for each random variable: the sample mean, used to estimate the expectation of the random variable, and the sample variance, used to measure the accuracy of that estimate. From these two properties measures of confidence may be generated.

<u>The Sample Mean and Sample Variance</u>. Let X be a random variable and X_i the value of X for the i^{th} Monte Carlo run. Then the sample mean $\overline{X}$ and the sample variance $\overline{\sigma}^2$ for m Monte Carlo runs are given by:

$$\overline{X} = \frac{1}{m} \sum_{i=1}^{m} X_i \tag{5-1}$$

$$\overline{\sigma}^2 = \frac{1}{m-1} \sum_{i=1}^{m} (X_i - \overline{X})^2 \tag{5-2}$$

Suppose X has mean μ and variance Σ. The central limit theorem implies that the distribution of $\overline{X}$ approaches normal with mean μ and variance Σ/m as m increases; furthermore, if $\overline{X}$ is distributed normally,

$$t = \frac{\overline{X} - \mu}{\sqrt{\overline{\sigma}^2/m}} \tag{5-3}$$

is distributed via Student's t with m degrees of freedom. But for sufficiently large m (say m > 100) Student's t can be replaced by the unit normal distribution (i.e., normal with mean zero, variance one). In the sequel it is assumed that m is sufficiently large such that t as defined in (5-3) is distributed with a unit normal distribution.

<u>Confidence Measures</u>. From the above it is clear that means of estimating μ for a given random variable X are desired and that $\overline{X}$ provides such an estimate. Naturally, the question of how good the estimate of μ is arises. Suppose t^* is selected such that Prob $(t \geq t^*) = 0.025$, where t has a unit normal distribution. Suppose that 100 Monte Carlo simulations are performed. Each simulation

may measure a different random variable; the only requirement is that a sufficient number of runs are made for each simulation to insure that t in (5-3) is unit normal. It may be expected that in 95 of these simulations that $-t^* \leq t \leq t^*$ or that

$$\bar{X} - t^* \sqrt{\sigma^2/m} \leq \mu \leq \bar{X} + t^* \sqrt{\sigma^2/m} \; . \qquad (5\text{-}4)$$

This approach to confidence was originated by E. S. Pearson[3]; the interval $[\bar{X} - t^* \sqrt{\sigma^2/m} \;,\; \bar{X} + t^* \sqrt{\sigma^2/m}\,]$ is known as the 95% two-sided confidence interval (other confidence intervals are defined in a similar manner). If every time a Monte Carlo simulation is made, a 95% confidence interval is used, then 95% of the time $\bar{X}$ will be in the 95% interval. However, note that for a given Monte Carlo run the probability that $\bar{X}$ lies in the interval is not 0.95--in fact this probability is not defined.[3]

An alternate approach to confidence was proposed by R. A. Fisher.[3] His approach is as follows. Assume that $\bar{X}$ and σ^2 are fixed at the values obtained in a Monte Carlo simulation and that t is unit normal; the distribution of t induces a distribution on μ known as the fiducial distribution. Now select μ_1 and μ_2 such that the fiducial probability that $\mu < \mu_1$ is 0.025 and the fiducial probability that $\mu > \mu_2$ is 0.025. The interval $[\mu_1, \mu_2]$ is known as the 95% fiducial interval; clearly in the present case $\mu_1 = \bar{X} - t^* \sqrt{\sigma^2/m}$ and $\mu_2 = \bar{X} + t^* \sqrt{\sigma^2/m}$ and the confidence and fiducial intervals are identical--this is not always the case however.[3]

Improving Monte Carlo Efficiency

Using the above techniques one can determine when enough Monte Carlo runs have been made to estimate any desired property to any desired accuracy as specified by the 95% (or any other %) confidence intervals. In the remainder of this section two methods are discussed for reducing the number of Monte Carlo runs required to achieve a given accuracy: the use of exact comparisons and the use of biased random numbers. The reader interested in pursuing this matter further is referred to reference 4.

Use of Exact Comparisons. Suppose that it is desired to compare two (or more) alternative Strategic C^2 networks and only differences in performance as opposed to absolute performance is of interest. For example, a baseline network may exist and an improved network may be proposed; it may then be desired to determine if the improved network yields a significant performance improvement compared to the baseline. Consider a single measure of performance. If separate Monte Carlo simulations are made for each network, the observed differences in performance between the two networks are due to two factors: the difference between the random numbers used in similar situations in each simulation, and the inherent difference in the capability of the two networks. Only the last factor is of interest. Differences in performance due to the first factor may be lessened as follows: let the same random numbers be chosen for the same event for the same Monte Carlo run in the simulation of each network, compare the two networks run by run, and find the mean and variance of the differences in performance. When two separate Monte Carlo runs are made for the two networks the variance of the difference in performance is the sum of the variances in performance for the two simulations. This variance is larger--and in general much larger--than the variance of the difference in performance obtained on a run-by-run comparison with identical random numbers; hence the number of Monte Carlo runs required to detect significant differences in performance between the two networks will in general be much fewer if the exact comparison technique is used. To perform an exact comparison using the Dynamic Network Simulator the random numbers used in each Monte Carlo run for node destruction, node delays, and link delays must be the same for corresponding nodes and links in the two networks (wherever such correspondence exists).

Use of Biased Random Numbers. Suppose that it is desired to see how a network will perform under stressing situations. Such situations may be emphasized in a Monte Carlo simulation by biasing the random numbers. Typically, when this is done more variations in behavior of the network (or other system elements being simulated) are obtained with the same number of Monte Carlo runs. As a result, a better distribution of outcomes is obtained with the same number of Monte Carlo runs (i.e., the distribution of t is more nearly normal). Biasing the random numbers in the Dynamic Network Simulator may be simply accomplished by picking random numbers with non-uniform distributions on the unit interval. (Longer node and link delays are obtained when the random numbers are larger; this may be done by squaring--or cubing, etc.--the random units and subtracting from 1, thus biasing the random number toward 1. Earlier node failures are obtained when the random numbers are smaller; this may be done by squaring the random units thus biasing the random number toward 0.)

When the random numbers are biased the computation of sample mean and sample variance must be modified to account for the fact that stressing situations are occurring more often than expected. The necessary modification is to use weighted averages in computing the sample mean and variance where the weighting for each Monte Carlo run is the ratio of its probability of occurrence in nature to the probability of its occurrence in the simulation. In the Dynamic Network Simulator this weighting factor is found as follows: let $r_1, \ldots r_m$ be all the random numbers required by the simulation, let $p_j(r)$ be the probability density for the i^{th} random number in the simulation, and let ζ_{ij} be the value of r_j used on the i^{th} run. Assuming r_j is a uniformly distributed random variable, the weighting factor ρ_i for the i^{th} run is

$$\rho_i = \prod_{j=1}^{m} 1/p_j(\zeta_{ij}) \tag{5-5}$$

The weighted sample mean and variance are:

$$\overline{X} = \frac{1}{m} \sum_{i=1}^{m} X_i \rho_i \tag{5-6}$$

$$\overline{\sigma}^2 = \frac{1}{m-1} \sum_{i=1}^{m} (X_i - \overline{X})^2 \rho_i \tag{5-7}$$

Note that if unbiased random numbers are used $\rho_i = 1$ and (5-6) and (5-7) reduce to (5-1) and (5-2).

VI. AN EXAMPLE

To make the previous discussion clearer, and to illustrate the type of outputs produced by the models, an example is presented in this section. It should be noted that this example is for illustrative purposes only; no conclusions should be drawn from the results.

Outputs of the Network Status Model

Consider a hypothetical network of 33 nodes and 97 links. On the basis of appropriate link, node, burst and jammer information, the Network Status Model generates outputs of the type given in Tables 6-1 through 6-6,* namely node status in terms of cumulative probability of destruction versus time, and link status in terms of either probability of availability versus time for VHF-EHF satellite, and special links, or probability of character error versus time for ELF-LF links. Tables 6-4 through 6-6 define the hypothetical network by giving the sending and receiving node for each link. If one inspects Tables 6-4 through 6-6 carefully one will note that no links either leave or enter nodes 2, 3, 13, and 26 through 32; these nodes are assumed to be inoperable for this example.

The Outputs of the Dynamic Network Simulator

The Dynamic Network Simulator requires for inputs the outputs of the Network Status Model such as those given in Tables 6-1 to 6-6. Assume for our hypothetical network that the outputs shown in Tables 6-1 to 6-6 have been generated by the Network Status Model and supplied to the Dynamic Network Simulator. Note that the special nodes, as described in Table 6-3, are divided into classes and have the values given in that table for receiving the message at a given time. The Dynamic Network Simulator produces a variety of outputs. For the example, illustrative outputs are shown in Figures 6-1 to 6-5 for 100 Monte Carlo runs. Figures 6-1 to 6-4 are time period histograms, taken at 60 minutes after EAM injection into the network, of the value of EAM receipt for the bomber nodes, the submarine nodes, the Minuteman nodes, and all special nodes. Figure 6-5

---

* The Network Status Model is under development -- the data shown in Tables 6-1 through 6-6 was made up for the example and not produced by the Network Status Model.

TABLE 6.1 PROBABILITY OF NODE DESTRUCTION FOR EXAMPLE

NODE	TIME	PK	TIME	PK	TIME	PK	TIME	PK
4	0	.20	10	.40	20	.60		
5	0	.20	10	.40	20	.60		
6	0	.20	10	.40	20	.60		
7	0	.20	10	.40	20	.60		
8	0	.10	10	.20	20	.30	30	.40
9	0	.10	10	.20	20	.30	30	.40
10	0	.10	10	.20	20	.30	30	.40
11	0	.20						
12	0	.20						
20	0	.10	20	.20				
21	0	.10	20	.20				
22	0	.50						
24	0	.20						

Notes:

1. PK is the cumulative probability that the node has been destroyed at the corresponding time or sooner.

2. Nodes not appearing in table are assumed to be indestructable.

TABLE 6.2 NODE DELAYS FOR EXAMPLE

NODE	MIN.	INT.	NODE	MIN.	INT.	NODE	MIN.	INT.	NODE	MIN.	INT.
1	0	0	2	0	0	3	0	0	4	3	5
5	3	5	6	3	5	7	3	5	8	3	5
9	3	5	10	3	5	11	3	5	12	3	5
13	0	0	14	0	0	15	0	0	16	0	0
17	0	0	18	0	0	19	0	0	20	5	3
21	5	3	22	0	0	23	0	0	24	3	5
25	0	0	26	0	0	27	0	0	28	0	0
29	0	0	30	0	0	31	0	0	32	0	0
33	0	0									

Min. is minimum value of node delay.

Int. is interval of node delay (i.e. maximum node delay = min. + int.)

TABLE 6.3 HIGH FREQUENCY LINKS IN EXAMPLE

LINK	N1	N2	CLASS	(TIME, PROBABILITY OF LINK AVAILABILITY)			
1	17	20	3	(0,.9)	(10,.3)	(20,.7)	(30,.4)
2	17	21	3	(0,.7)			
3	1	22	2	(0,.6)	(10,.9)	(20,.1)	(30,.5)
4	22	24	2	(0,.7)			
5	22	7	2	(0,.7)			
6	22	4	2	(0,.7)			
7	22	6	2	(0,.7)			
8	22	5	2	(0,.7)			
9	1	12	3	(0,.7)			
10	12	11	3	(0,.7)			
11	11	7	3	(0,.7)			
12	7	4	3	(0,.7)			
13	4	7	3	(0,.7)			
14	4	6	3	(0,.7)			
15	6	4	3	(0,.7)			
16	6	7	3	(0,.7)			
17	7	6	3	(0,.7)			
18	7	17	3	(0,.7)			
19	7	17	3	(0,.7)			
20	4	5	3	(0,.7)			
21	5	4	3	(0,.7)			
22	5	6	3	(0,.7)			
23	6	5	3	(0,.7)			
24	5	8	3	(0,.7)			
25	8	5	3	(0,.7)			
26	5	18	3	(0,.6)	(10,.9)	(20,.1)	(30,.5)
27	8	6	3	(0,.7)			
28	6	8	3	(0,.7)			
29	8	15	3	(0,.7)			
30	8	9	3	(0,.7)			
31	9	8	3	(0,.7)			
32	9	14	3	(0,.9)	(10,.3)	(20,.7)	(30,.4)
33	8	10	3	(0,.7)			
34	10	8	3	(0,.7)			
35	10	6	3	(0,.7)			
36	6	10	3	(0,.7)			
37	10	19	3	(0,.7)			
38	10	16	3	(0,.7)			

Notes:

1. N1 is sending node; N2 is receiving node.
2. Class 2 links are satellite links.
3. Link probability of availability is assumed to hold from the corresponding time to the next larger time.

TABLE 6.4 LOW FREQUENCY LINKS IN EXAMPLE

LINK	N1	N2	MESS	(TIME, PROBABILITY OF CHARACTER ERROR)
39	1	16	13	(0,.050) (5,.040) (10,.030) (15,.020) (20,.010) (25,.001)
40	1	19	13	(0,.020) (5,.040) (10,.050) (15,.040) (20,.030) (25,.020) (30,.010) (35,.005) (40,.005)
41	1	6	13	(0,.001) (5,.020) (10,.030) (15,.030) (20,.010) (25,.020) (30,.001)
42	1	7	13	(0,.050) (5,.040) (10,.030) (15,.020) (20,.010) (25,.001)
43	1	5	13	(0,.020) (5,.040) (10,.050) (15,.040) (20,.030) (25,.020) (30,.010) (35,.005) (40,.005)
44	1	4	13	(0,.001) (5,.020) (10,.030) (15,.030) (20,.010) (25,.020) (30,.001)
45	1	17	13	(0,.050) (5,.040) (10,.030) (15,.020) (20,.010) (25,.001)
46	1	18	13	(0,.020) (5,.040) (10,.030) (15,.040) (20,.030) (25,.020) (30,.010) (35,.005) (40,.005)
47	1	15	13	(0,.001) (5,.020) (10,.030) (15,.030) (20,.010) (25,.020) (30,.001)
48	7	16	13	(0,.050) (5,.040) (10,.030) (15,.020) (20,.010) (25,.001)
49	7	19	13	(0,.020) (5,.040) (10,.050) (15,.040) (20,.030) (25,.020) (30,.010) (35,.005) (40,.005)
50	7	14	13	(0,.001) (5,.020) (10,.030) (15,.030) (20,.010) (25,.020) (30,.001)
51	7	15	13	(0,.050) (5,.040) (10,.030) (15,.020) (20,.010) (25,.001)
52	7	18	13	(0,.020) (5,.040) (10,.050) (15,.040) (20,.030) (25,.020) (30,.010) (35,.005) (40,.005)
53	7	17	13	(0,.001) (5,.020) (10,.030) (15,.030) (20,.010) (25,.020) (30,.001)
54	7	6	13	(0,.050) (5,.040) (10,.030) (15,.020) (20,.010) (25,.001)
55	7	5	13	(0,.020) (5,.040) (10,.050) (15,.040) (20,.030) (25,.020) (30,.010) (35,.005) (40,.005)
56	7	4	13	(0,.001) (5,.020) (10,.030) (15,.030) (20,.010) (25,.020) (30,.001)
57	6	7	13	(0,.050) (5,.040) (10,.030) (15,.020) (20,.010) (25,.001)
58	6	16	13	(0,.020) (5,.040) (10,.050) (15,.040) (20,.030) (25,.020) (30,.010) (35,.005) (40,.005)
59	6	19	13	(0,.001) (5,.020) (10,.030) (15,.030) (20,.010) (25,.020) (30,.001)
60	6	14	13	(0,.050) (5,.040) (10,.030) (15,.020) (20,.010) (25,.001)
61	6	15	13	(0,.020) (5,.040) (10,.050) (15,.040) (20,.030) (25,.040) (30,.020) (35,.005) (40,.005)
62	6	18	13	(0,.001) (5,.020) (10,.030) (15,.030) (20,.010) (25,.020) (30,.001)
63	6	5	13	(0,.050) (5,.040) (10,.030) (15,.020) (20,.010) (25,.001)

TABLE 6.4 LOW FREQUENCY LINKS IN EXAMPLE (CONT.)

LINK	N1	N2	MESS	(TIME, PROBABILITY OF CHARACTER ERROR)
64	6	17	13	(0,.020) (5,.040) (10,.050) (15,.040) (20,.030) (25,.020) (30,.010) (35,.005) (40,.005)
65	6	4	13	(0,.001) (5,.020) (10,.030) (15,.030) (20,.010) (25,.020) (30,.001)
66	4	7	13	(0,.050) (5,.040) (10,.030) (15,.020) (20,.010) (25,.001)
67	4	6	13	(0,.020) (5,.040) (10,.050) (15,.040) (20,.030) (25,.020) (30,.010) (35,.005) (40,.005)
68	4	19	13	(0,.001) (5,.020) (10,.030) (15,.030) (20,.010) (25,.020) (30,.001)
69	4	16	13	(0,.050) (5,.040) (10,.030) (15,.020) (20,.010) (25,.001)
70	4	14	13	(0,.020) (5,.040) (10,.050) (15,.040) (20,.030) (25,.020) (30,.010) (35,.005) (40,.005)
71	4	15	13	(0,.001) (5,.020) (10,.030) (15,.030) (20,.030) (25,.020) (30,.001)
72	4	18	13	(0,.050) (5,.040) (10,.030) (15,.020) (20,.010) (25,.001)
73	4	17	13	(0,.020) (5,.040) (10,.050) (15,.040) (20,.020) (25,.020) (30,.010) (35,.005) (40,.005)
74	5	6	13	(0,.001) (5,.020) (10,.030) (15,.030) (20,.010) (25,.020) (30,.001)
75	5	7	13	(0,.050) (5,.040) (10,.030) (15,.020) (20,.010) (25,.001)
76	5	4	13	(0,.020) (5,.040) (10,.050) (15,.040) (20,.020) (25,.020) (30,.010) (35,.005) (40,.005)
77	5	17	13	(0,.001) (5,.020) (10,.030) (15,.060) (20,.010) (25,.020) (30,.001)
78	5	18	13	(0,.050) (5,.040) (10,.030) (15,.020) (20,.010) (25,.001)
79	5	15	13	(0,.020) (5,.040) (10,.050) (15,.040) (20,.020) (25,.020) (30,.010) (35,.005) (40,.005)
80	5	14	13	(0,.050) (5,.040) (10,.030) (15,.020) (20,.010) (25,.001)
81	5	16	13	(0,.001) (5,.020) (10,.030) (15,.030) (20,.010) (25,.020) (30,.001)
82	5	19	13	(0,.050) (5,.040) (10,.030) (15,.020) (20,.010) (25,.001)
83	4	5	13	(0,.040) (5,.040) (10,.030) (15,.020) (20,.010) (25,.001)
84	24	33	13	(0,.020) (5,.030) (10,.050) (15,.040) (20,.020) (25,.020) (30,.010) (35,.005) (40,.005)

Notes:

1. N1 is sending node; N2 is receiving node.
2. All low frequency links are in class 4.
3. Link probability of character error is assumed to hold from corresponding time to next larger time.
4. "MESS" is the time required for a single transmittal of the message.

TABLE 6.5 SPECIAL LINKS IN EXAMPLE

LINK	N1	N2	TIME DELAY	(TIME, PROBABILITY OF AVAILABILITY)
85	20	19	4	(0,.7)
86	20	16	4	(0,.7)
87	20	14	4	(0,.9) (10,.3) (20,.7) (30,.4)
88	20	15	4	(0,.7)
89	20	18	4	(0,.7)
90	20	25	15	(0,.7)
91	21	23	15	(0,.7)
92	21	24	15	(0,.7)
93	21	19	4	(0,.7)
94	21	16	4	(0,.7)
95	21	14	4	(0,.7)
96	21	15	4	(0,.7)
97	21	18	4	(0,.6) (10,.9) (20,.1) (30,.5)

1. N1 is sending node; N2 is receiving node.
2. All special links are class 1 links.
3. Time Delay is the time delay after EAM arrival before the link becomes available.
4. Link probability is assumed to hold from the corresponding time to the next larger time.

TABLE 6.6 SPECIAL NODES AND THEIR PROPERTIES

NODE	CLS	(TIME, VALUE) PAIRS			
23	1	(0,499.00)	(10,450.00)	(20,375.00)	(30,250.00)
25	1	(0,499.00)	(10,450.00)	(20,375.00)	(30,250.00)
33	2	(0,999.00)			
14	3	(0,200.00)	(10,100.00)	(20,40.00)	(30,30.00)
15	3	(0,150.00)	(10,70.00)	(15,40.00)	(20,30.00)
16	3	(0,200.00)	(10,100.00)	(20,40.00)	(30,30.00)
17	3	(0,150.00)	(10,70.00)	(15,40.00)	(20,30.00)
18	3	(0,200.00)	(10,100.00)	(20,40.00)	(30,30.00)
19	3	(0,150.00)	(10,70.00)	(15,40.00)	(20,30.00)

Notes:

1. Class 1 nodes are bomber nodes; class 2 nodes are submarine nodes; class 3 nodes are Minuteman nodes.
2. Value of the EAM to a node is assumed constant from the associated time to the next larger time.

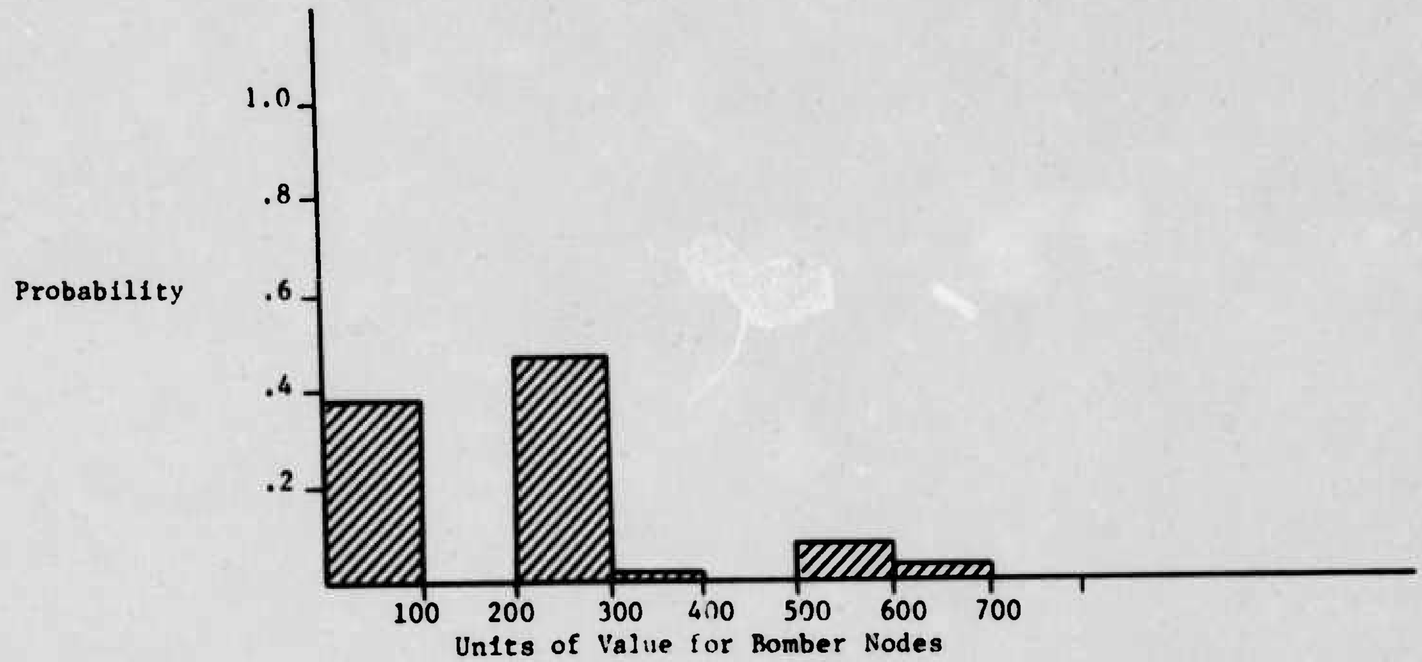


FIGURE 6.1 TIME PERIOD HISTOGRAM OF VALUE ACCRUED BY BOMBER NODES FOR HAVING RECEIVED THE EAM AT OR PRIOR TO t=60 MINUTES

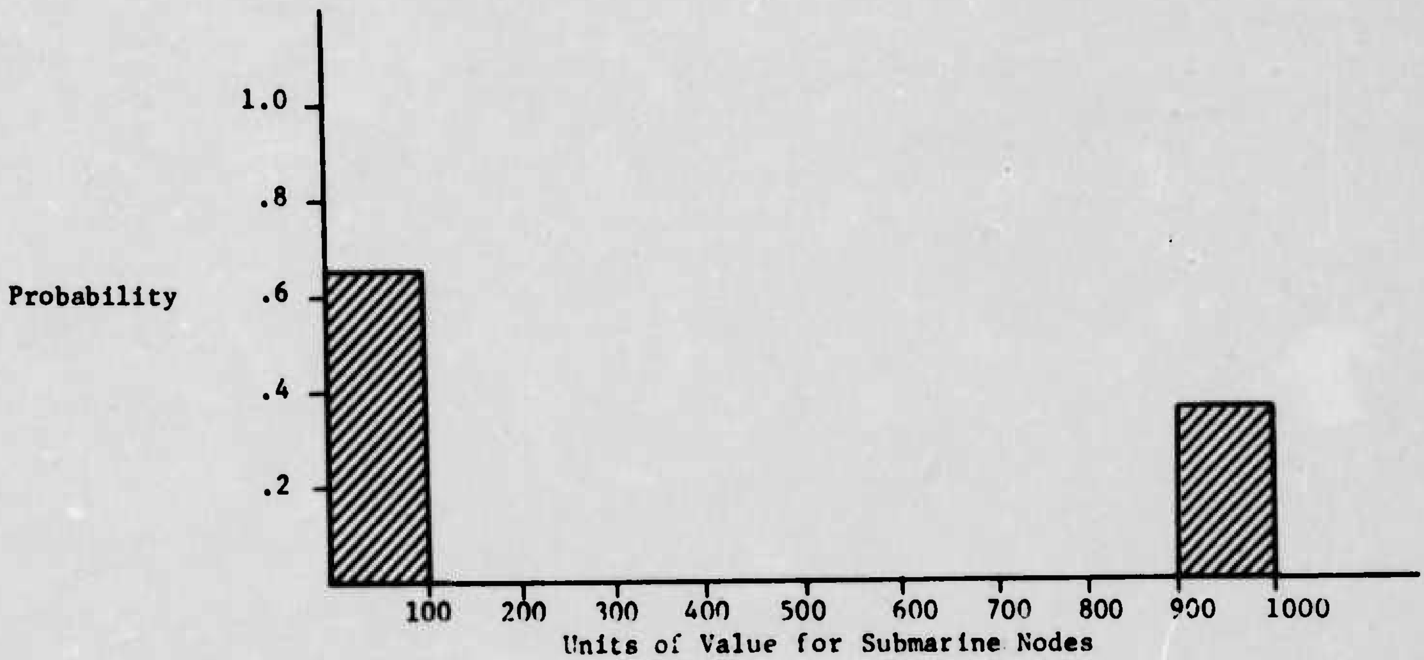


FIGURE 6.2 TIME PERIOD HISTOGRAM OF VALUE ACCRUED BY SUBMARINE NODES FOR HAVING RECEIVED THE EAM AT OR PRIOR TO t=60 MINUTES

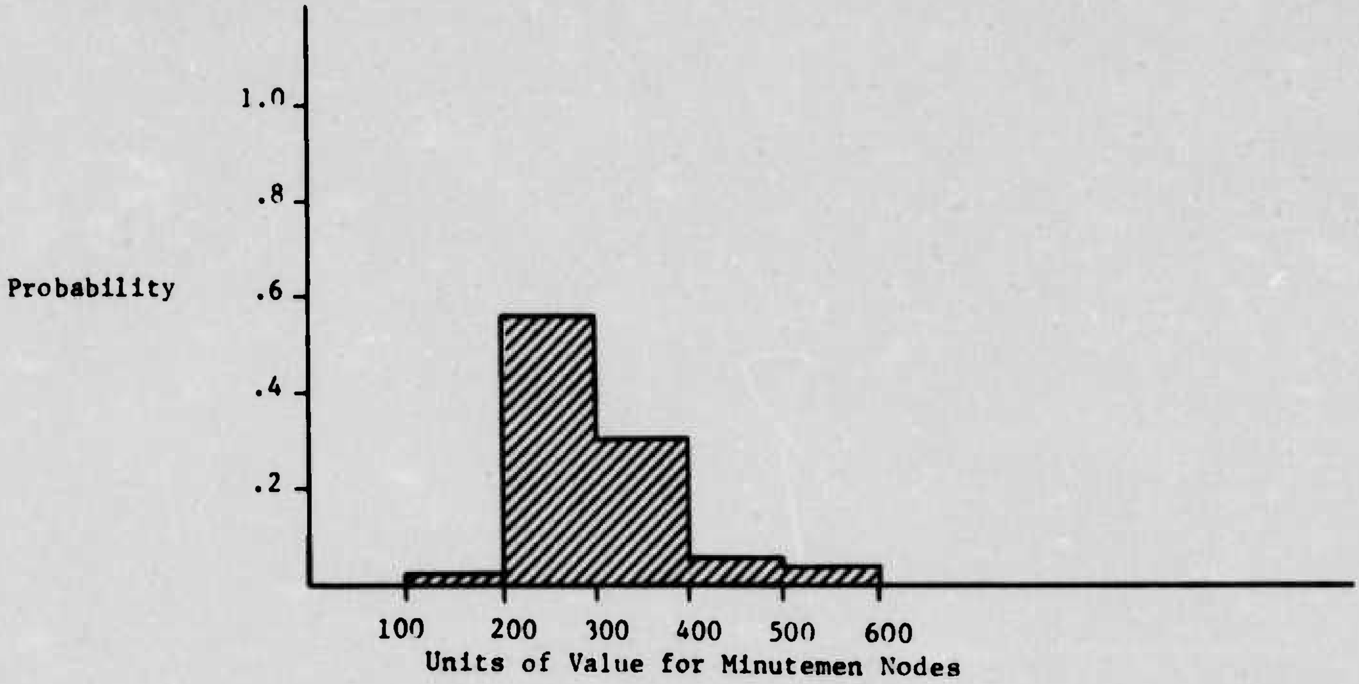


FIGURE 6.3 TIME PERIOD HISTOGRAM OF VALUE ACCRUED BY MINUTEMAN NODES FOR HAVING RECEIVED THE EAM AT OR PRIOR TO t=60 MINUTES

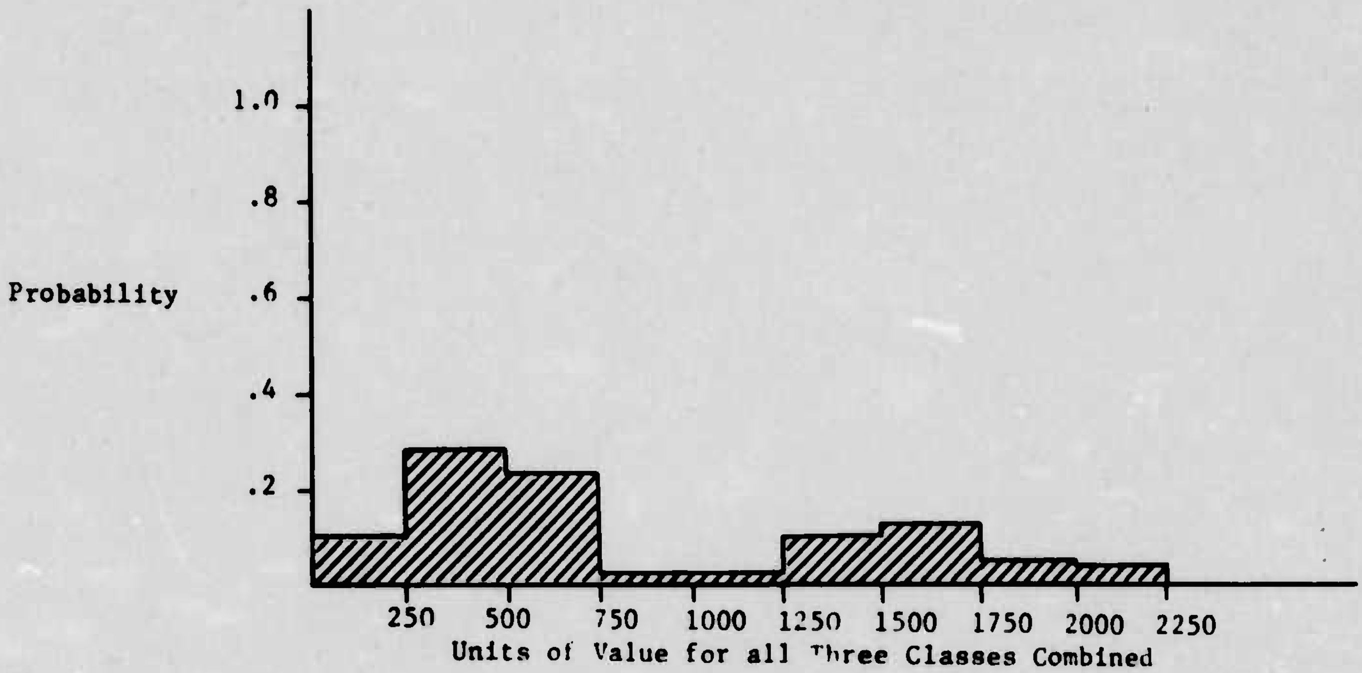
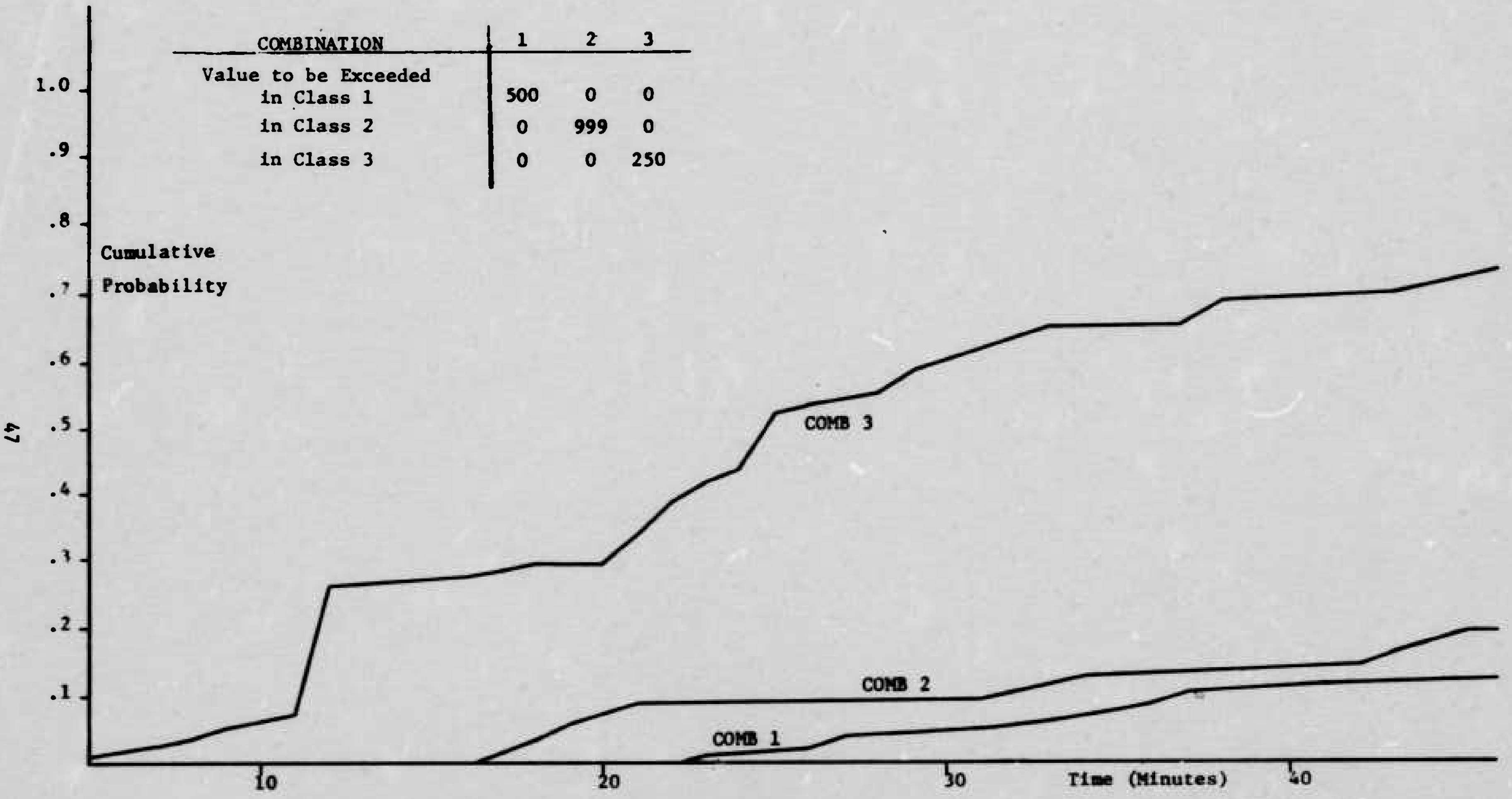


FIGURE 6.4 TIME PERIOD HISTOGRAM OF VALUE ACCRUED BY ALL SPECIAL NODES FOR HAVING RECEIVED THE EAM AT OR PRIOR TO t=60 MINUTES



COMBINATION	1	2	3
Value to be Exceeded in Class 1	500	0	0
in Class 2	0	999	0
in Class 3	0	0	250

FIGURE 6.5 CUMULATIVE PROBABILITY VS TIME THAT CERTAIN COMBINATIONS OF VALUE ARE EXCEEDED

plots the cumulative probability that certain combinations of value are exceeded versus time.

Suppose that the final time of interest is 60 minutes (i.e., value of receiving the message after that time is zero). Then Figure 6-4 is a histogram of the performance index (i.e., the total value of the nodes that receive the EAM). The expected performance is $J = 860$.

VII. EXTENSIONS OF THE PRESENT MODEL

This report has described two models -- the Network Status Model and the Dynamic Network Simulator that can be used to simulate in an efficient manner the behavior of a Strategic C^2 network delivering attack orders. The Network Status Model should be applicable with little change to other Strategic C^2 problems such as the Attack Warning and Battle Management problems. On the other hand, the Dynamic Network Simulator in its present form is applicable only to delivery of attack orders. The heart of the Dynamic Network Simulator is a version of the shortest path algorithm. Direct use of the SPA is possible because it has been assumed that only one message is transmitted by the network and that message is transmitted to all nodes. In the Attack Warning and Battle Management problems more than one message is sent over the network and each message is sent to only one of a few nodes. For these problems routing and priority of messages become important and the DNS must be modified to simulate the behavior of the Strategic C^2 Network employed.

The Shortest Path Algorithm Viewed as a Discrete Simulation

In order to extend the Dynamic Network Simulator to address the Attack Warning and Battle Management problems it is helpful to view it from a different point of view. A discrete event simulation is one in which events are time ordered and performed in time sequence; any event may create new future events, cancel future events, or change the time of future events. The basis of a discrete event simulation is the event list containing the events to be performed; the simulation proceeds by performing the earliest event on the list. Now, if events are defined to be arrival of the message at a node then it is clear that the SPA as used in the Dynamic Network Simulator is a discrete event simulation of a Strategic C^2 Network and that the consideration list is the event list for the simulation.

Communication networks have been modelled by discrete event simulations in the past (see for example Ref. 5); however, the simulation in Ref. 5 required significantly more computer time than that of the simulation described in this report.[5] The reasons for the relative computer time requirements of the two simulations are threefold; computers are now faster, the previous simulation used more than the minimum number of events that are necessary, and the previous simulation did not use the efficient organization provided by the SPA. The latter two reasons relate to the efficiency of processing the event list, which in turn depends upon how the event list is ordered. If the event list contains events listed in order of their creation then it must be searched each time an event is created, deleted or modified. Such searches will typically be linear -- hence the search time is proportional to the length of the list. However, if a binary list is used a binary search (which has search time proportional to the logarithm of list length) is possible when placing new events in the list but not when deleting or modifying existing events on the list.

A discrete event simulation based upon the SPA as described in this report is superior to standard discrete event simulations in two respects: First, message arrivals are the only events considered; any event -- such as weapon launch -- that is dependent on message arrival can readily be modelled as part of the message arrival event, and thus the number of events that must be considered is minimized. Secondly, by quantizing time and using the node list, the consideration list need not be completely searched. To see this, view the consideration list as a two dimensional array where one dimension is quantized time and the second dimension is all events that occur at that time in any order (e.g., all nodes that receive a message at that time in any order). It is clear that if the information contained in the node list is used, creating, modifying or deleting an event involves only a search on the second dimension of this two dimensional array. Since the number of nodes receiving a message at a given time is small, the required search time is small. The major drawback to this approach is the excessive storage that would be required if very small time quantization levels were required.

Message Routing and Priority

The discrete event simulation developed for the Delivery of Attack Orders Problem may be readily modified to treat the Attack Warning and Battle Management Problems. When so modified, its heart is no longer a SPA but it retains the list processing efficiency of the SPA. The major required modifications relate to incorporation of message priority and routing logic. The capacity of links is limited to a small number of messages at one time over a high frequency link and one message at a time over a low frequency link. If more messages must be sent over a link than its capacity allows, the sequence in which messages are sent must be determined. For a given message, if no direct link exists between the node under consideration and the ultimate destination of the message, then the next node to which the message is sent must be selected. Generation of the logic for routing messages is a task that is made no easier by the fact that the sender may be unaware of the status of the links he has available for sending messages. An important use of a modified Dynamic Network Simulator would be in comparing the efficiency of various existing and proposed routing logics.

APPENDIX A. NODE DESTRUCTION ALGORITHM

For nodes located on the ground and for ground level nuclear bursts, the node kill probabilities are computed by a standard nuclear weapons effect algorithm[6] in the Network Status Model. This model sets the probability of destruction of the node equal to the circular coverage function $P(R,r,\sigma)$ which is defined to be:

$$P(R,r,\sigma) = \int_D N(x,y)\,dx\,dy,$$

where

$N(x,y)$ = 2-dimensional normal distribution function for mean 0 and covariance 1,

$$D = \left\{ (x,y):(u-x)^2 + (v-y)^2 < (R/\sigma)^2,\ u^2 + v^2 = (r/\sigma)^2 \right\}.$$

Note that from the axial symmetry of $N(x,y)$, the integral is independent of the value of u and v chosen provided $u^2 + v^2 = (r/\sigma)^2$.

Very briefly the model is as follows. The parameter r is taken to be the ground distance between the node and the burst, while R is computed from the VN hardness of the target and the warhead size. The parameter σ is found from

$$\sigma^2 = \sigma_D^2 + \sigma_E^2$$

where σ_D is a known function of R which depends upon whether the target is overpressure or dynamic pressure sensitive (indicated in the VN number by a P or Q), and σ_E is given by

$$\sigma_E = \text{(CEP of warhead)}\ /1.1774$$

It is assumed that nodes are point targets; if this is not true a term σ_V^2 must be added to the equation of σ^2 in order to reflect the size of the node.

The computation of R proceeds as follows. From the VN number and warhead size, the overpressure or dynamic pressure at which the node will be killed 50% of the time is computed from known equations.[6] R is the ground radius from the burst point at which this overpressure or dynamic pressure occurs. In the model

R is found from an approximate equation fitted to tabular data. Two fits are possible: one for ground level bursts and one for bursts at optimum height above ground. The input nuclear weapon laydown tape must indicate for each ground burst whether it is an optimum or zero height burst.

APPENDIX B. USE OF SHORTEST PATH ALGORITHM TO DETERMINE STATIC CONNECTIVITY

The shortest path algorithm presented in the body of this report may be used to predict static connectivity of source to sink nodes for networks with random failures of nodes and links. Let

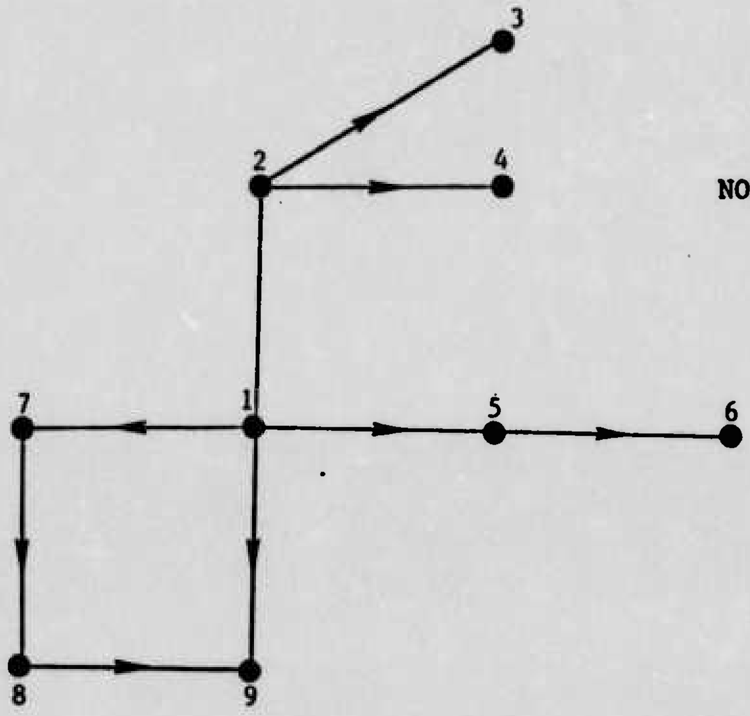
q_i = probability node i is not operable at some specified time

P_{ij} = probability link ij is not operable at some specified time.

Before each call of the shortest path algorithm during each Monte Carlo run, a random unit r_i is selected for each node i. If $r_i < q_i$ then the time of node destruction t_i^d is set to 0; otherwise $t_i^d = \infty$. Each time a link is considered in the shortest path algorithm, a random unit r_{ij} is selected. If $r_{ij} < P_{ij}$ then the link delay d_{ij} is set to ∞; otherwise $d_{ij} = 0$. Nodes with $k^*(i) = 0$ at the end of each Monte Carlo run are connected to the source; nodes with $k^*(i) = \infty$ are not connected to the source.

Two examples were run to compare the shortest path algorithm with a standard connectivity algorithm (SCA)[1]. The example illustrated in Fig. B.1 is sufficiently simple that the connectivities with the source node may be easily computed by hand -- these values are given in Table B.1. Two different results are given for nodes 7 and 8 depending upon whether links 7-8 and 8-9 are directed or not. If p is the probability of being disconnected, then the standard deviation in a Monte Carlo determination of p is $p(1-p)/N$ where N is the number of Monte Carlo runs. Computed standard deviations for N = 900 are also given in Table B.1. Monte Carlo determinations of p for N = 900 from both the standard connectivity algorithm (SCA) and shortest path algorithm (SPA) are given in Table B.1, as well as the errors in these estimates normalized by the appropriate standard deviations.

Both sets of answers closely approximate the true probabilities of disconnection. The average standard deviations are biased negatively, but this is probably caused by the correlation between connectivities as the following



NODE FAILURE PROBABILITIES

i	q_i
1	0.2
2	0.1
3	0.0
4	0.2
5	0.1
6	0.0
7	0.2
8	0.1
9	0.0

LINK FAILURE PROBABILITIES

= 0.2 FOR ALL LINKS

FIG. B-1 AN EXAMPLE NETWORK

TABLE B-1

RESULTS FOR EXAMPLE OF FIG. B-1

j	PROBABILITY OF DISCONNECTION*	STANDARD DEVIATION	ESTIMATED PROBABILITY OF DISCONNECTION SCA	SPA	ERROR/STANDARD DEVIATION SCA	SPA
2	0.424	0.016	0.400	0.403	-1.5	-1.31
3	0.539	0.017	0.490	0.507	-2.59	-1.88
4	0.631	0.016	0.632	0.624	0.06	-0.50
5	0.424	0.016	0.399	0.422	-1.57	-0.12
6	0.539	0.017	0.525	0.531	-0.80	-0.47
7	0.429	0.016	0.387		-2.65	
7**	0.488	0.016		0.474		-0.87
8	0.406	0.016	0.377		-1.86	
8**	0.631	0.016		0.632		+0.06
9	0.301	0.015	0.281	0.270	-1.33	-2.07
				Ave.	-1.53	-0.87

* Disconnection here implies Node 1 and Node j.

** Link Directed.

analysis shows. In the SPA Monte Carlo runs it was found that the source node failed 19.1% of the time as compared to a computed value of 20.0%. The normalized error is -0.69 standard deviations; however, since failure of the source node implies no connectivity, this result biases all the connectivities on the negative side.

A more complicated example shown in Table B.2 was considered. In this example there are 34 nodes (only 31 of which are operable), each with a probability of 0.1 of failure, and 91 links, each with a probability 0.2 of failure. Both the SCA and the SPA programs were used to determine the probabilities that nodes were connected with the source node by making 100 Monte Carlo runs; results are given in Table B.3, in which X_1, X_2, and X_3 are the SCA (undirected), SPA undirected, and SPA directed probabilities of connection. (With the SPA program connectivity was computed both for links directed as in Table B.2 and for bi-directional links by assuming a uni-directional link each way for each link of Table B.2.) Standard deviations for the network with bi-directional links were computed by the SCA program. Table B.3 also gives the error between the bi-directional results from the two programs normalized by $\sqrt{2}$ times the standard deviation (because each is in error from the true value with a covariance equal to the standard deviation squared) Again the average normalized difference is biased, largely because in the SCA program the source node 1 failed only 9 times and in the SPA program it failed only 4 times instead of the expected 10 times. To remove this effect

$$X_4 = \frac{0.90}{0.91} X_1, \quad X_5 = \frac{0.90}{0.96} X_2$$

were computed; the results as well as the normalized differences between X_4 and X_5 are displayed in Table B.3. The average normalized difference between the two for bi-directional links is now 0.07 standard deviations, indicating that two programs give essentially the same results. The time taken on a UNIVAC 1108 computer to obtain the results of Table B.3 via the SPA was 1.5 seconds to find the connectivities with directed links and 2.8 seconds to find the connectivities with undirected links.

TABLE B.2 AN EXAMPLE NETWORK

LINK	N1	N2	LINK	N1	N2	LINK	N1	N2	LINK	N1	N2
1	1	12	26	1	26	51	18	34	76	5	17
2	7	11	27	4	27	52	19	34	77	5	18
3	5	8	28	5	28	53	1	4	78	5	19
4	6	8	29	6	24	54	1	5	79	6	7
5	6	10	30	7	30	55	1	6	80	6	14
6	8	9	31	25	31	56	1	7	81	6	15
7	8	10	32	23	32	57	1	15	82	6	16
8	8	15	33	1	22	58	1	16	83	6	17
9	9	14	34	22	4	59	1	17	84	6	18
10	10	16	35	22	5	60	1	18	85	6	19
11	10	19	36	22	6	61	1	19	86	7	14
12	11	12	37	22	7	62	4	5	87	7	15
13	17	20	38	22	24	63	4	6	88	7	16
14	17	21	39	24	33	64	4	7	89	7	17
15	20	25	40	26	34	65	4	14	90	7	18
16	21	23	41	27	34	66	4	15	91	7	19
17	21	24	42	28	34	67	4	16			
18	4	5	43	29	34	68	4	17			
19	4	6	44	30	34	69	4	18			
20	4	7	45	31	34	70	4	19			
21	4	17	46	32	34	71	5	6			
22	5	6	47	14	34	72	5	7			
23	5	18	48	15	34	73	5	14			
24	6	7	49	16	34	74	5	15			
25	7	17	50	17	34	75	5	16			

Note: If link is directed N1 is sending node and N2 is receiving node.

TABLE B.3

RESULTS FOR EXAMPLE OF TABLE B-2

	X_1 SCA	X_2 SPA UNDIRECTED	X_3 SPA DIRECTED	SCA ST. DEV	$\frac{X_1-X_2}{\sqrt{2}\,X_3}$	X_4 CORRECTED SCA	X_5 CORRECTED SPA	$\frac{X_4-X_5}{\sqrt{2}\,X_3}$
1	0.91	0.96	0.88	0.03	-1.2	0.90	0.90	
4	0.85	0.86	0.71	0.04	-0.2	0.84	0.81	+0.5
5	0.87	0.88	0.79	0.03	-0.2	0.86	0.83	+0.7
6	0.84	0.93	0.78	0.04	-1.6	0.83	0.87	-0.7
7	0.84	0.82	0.81	0.04	+0.4	0.83	0.77	+1.1
8	0.83	0.85	0.71	0.04	-0.4	0.82	0.80	+0.4
9	0.75	0.82	0.49	0.04	-1.2	0.74	0.77	-0.5
10	0.83	0.87	0.67	0.04	-0.7	0.82	0.82	0.0
11	0.68	0.72	0.55	0.05	-0.6	0.67	0.67	0.0
12	0.71	0.78	0.74	0.05	-1.0	0.70	0.73	+0.4
14	0.81	0.89	0.80	0.04	-1.4	0.80	0.83	-0.5
15	0.79	0.91	0.78	0.04	-2.1	0.78	0.85	-1.2
16	0.82	0.84	0.78	0.04	-0.4	0.81	0.79	+0.4
17	0.83	0.89	0.79	0.04	-1.1	0.82	0.83	-0.2
18	0.85	0.84	0.81	0.04	+0.2	0.84	0.79	+0.9
19	0.84	0.87	0.81	0.04	-0.5	0.83	0.82	+0.2
20	0.68	0.71	0.52	0.05	-0.4	0.67	0.67	0.0
21	0.74	0.78	0.57	0.04	-0.7	0.73	0.73	0.0
22	0.85	0.90	0.63	0.04	-0.9	0.84	0.84	0.0
23	0.65	0.72	0.43	0.05	-1.0	0.64	0.67	-0.2
24	0.74	0.82	0.61	0.04	-1.4	0.73	0.77	-0.7
25	0.63	0.67	0.40	0.05	-0.3	0.62	0.63	-0.1
26	0.79	0.79	0.65	0.04	0.0	0.78	0.74	+0.7
27	0.81	0.83	0.48	0.04	-0.4	0.80	0.78	+0.4
28	0.82	0.78	0.61	0.04	+0.7	0.81	0.73	+1.4
29	0.77	0.86	0.56	0.04	-1.6	0.76	0.81	-0.9
30	0.78	0.86	0.58	0.04	-1.4	0.77	0.81	-0.7
31	0.73	0.72	0.29	0.04	+0.2	0.72	0.67	+0.7
32	0.66	0.72	0.28	0.05	-0.9	0.65	0.67	-0.3
33	0.56	0.60	0.49	0.05	-0.6	0.55	0.56	-0.1
34	0.85	0.87	0.80	0.04	-0.4	0.84	0.82	+0.4
				AVE	-0.67			+.07

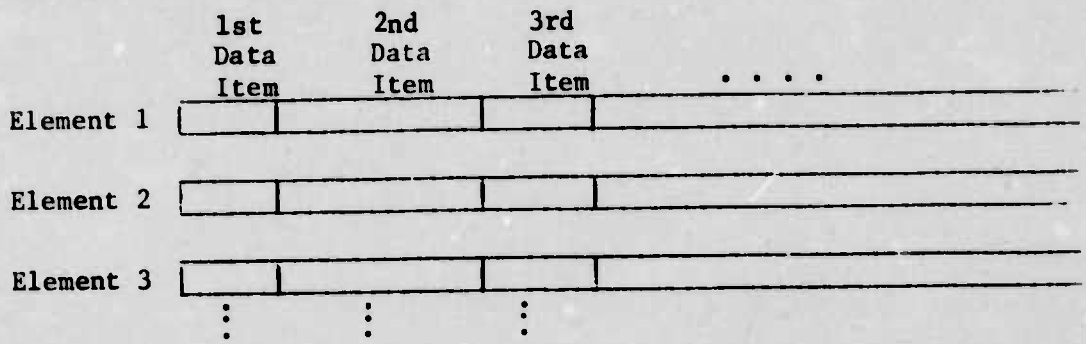
APPENDIX C. ORGANIZING THE DATA BASE

In typical Strategic C^2 networks of interest for analysis there could be as many as 700 links and 300 nodes. Each of these elements has its own particular set of characteristics as well as a generic set of characteristics representative of its element type. This represents a sizable data base that must be properly organized and managed as a first step in understanding and improving a Strategic C^2 system as a whole. The need for such structuring becomes even more apparent when trying to assess the effectiveness of a number of postulated improvements in a network and its components. The objective of organizing the Strategic C^2 data base to facilitate effectiveness studies may be met by a computerized data management and information retrieval system. Development of this system would require the accomplishment of three identifiable sub-tasks: designing suitable formats for data storage on various media; creating the software for data input, retrieval, and editing; and obtaining the baseline data and implementing the system. Thereafter the data base must be maintained in a useful state by purging outdated entries, adding new entries, and modifying spurious entries.

Data Requirement

Computer processable records may be distinguished on the basis of their length attribute: fixed or variable. The length, is measured in bytes (which is synonomous with characters in this context), and 'variable' implies that each record in the set may have a different length while 'fixed' implies that each record has the same length. Though variable records offer certain economies of storage they are clumsy to manage in data retrieval and sorting applications; for this reason fixed records are preferred in the present application. Each fixed record may be subdivided into fields of contiguous strings of bytes with fixed starting positions to record individual data items. The length of these fields may vary within a record, but is the same for all records within the data set. Since the field must contain values of variables, its length must be adequate to encompass the entire range of variation of the variable to which it is assigned. Thus four bytes must be provided to record frequencies in the range 0000 to 1000 MHZ with a resolution of 1 MHz.

The most convenient organization of the C^2 data base is to generate two data sets of fixed records; one data set being allocated to link data and the other to node data. A record in either data set can then be used to record all pertinent information about a particular link or node. Schematically each data set would then have the following organization:



The ordering of data items within records is less important than a determination of what those items should be. This determination must be made on the basis of present and future availability of data balanced against a judgement of what data is desired; thus a two-pronged effort is indicated. Ideally, elements of the C^2 system must be analyzed to extract the relevant characteristics of each element that affect overall system performance, regardless of whether or not this data is available. On the other hand, a literature survey must be conducted to ascertain the actual and projected availability of data.

An effort such as that outlined above can isolate the generic data items, but suffers from a lack of specificity with regard to particular elements. That is, fields within the record must be reserved for status information such as: a flag to indicate if that element is operative; indicators of significant departures from operating norms for that type of element; etc. Equally important is inter-connectivity information -- nodes for example must have pointers that indicate other nodes that address it, as well as pointers to nodes that it addresses. The links over which this communication occurs must also be recorded as pointers from the node data set to the link data set and vice versa.

Software Requirements

All of the data discussed above has the added dimension of temporal variation: nodes and links are added or deleted from the C^2 system, link characteristics may suffer diurnal and annual variation, nodes are upgraded by hardware modifications and downgraded by hardware malfunctions. These variations imply that fast, convenient means for modifying the data base and for displaying status information must be sought. This in turn outlines the requirements for several desirable computer programs which are discussed below.

Data Input and Validation Program

There is a significant step between possessing pertinent data and transcribing it on its ultimate storage medium (tape, disc, or magnetic drum) in the required format. Normally, printed data is transcribed on punched cards that are read by computer and transferred to the storage device. This error prone process is complicated by the necessity of using other than the desired final format on the punched cards. The effect of keypunching errors may be minimized by designing a simplified format for the punched cards, and using a program to reformat the data and validate it at the same time. For example, the validation program could reject an entry for a node having a transmit frequency of 10 Hz connected to a tropospheric scatter link.

Data Base Update Program

This program would provide three facilities and operate on the stored data base:

1) Modification - a facility to correct spurious entries in the data base, or to alter data items that have changed since the last updating period.

2) Insertion - a facility for adding new elements or records to the existing data sets.

3) Deletion - a facility for deleting elements or records from the existing data sets.

This program must be designed with a simplified language for performing its various functions so that a minimum of training would be required to use it. At the same time, it must have built in safeguards to prevent inadvertent destruction of valuable data. At the very least this implies that the update program must provide an automatic backup capability to preserve the data sets in unedited form.

Data Retrieval and Display Programs

This may be one or a collection of programs to interface between the storage media and the data user. An interactive system, perhaps utilizing a cathode ray tube, would be highly desirable. The system would provide a simplified retrieval language enabling the user to specify his needs. Computer prompting could be incorporated to lead the user through a decision tree of possible data alternatives. These programs must have the ability to sort the data on various keys, retrieve specific data items, and generate simple statistics. Some example commands that this system could support are:

DISPLAY STATUS OF LINK 230
LIST INPUTS TO NODE 570
GIVE PERCENT OF OPERATIONAL VLF LINKS
LIST ALL SEQUENCES OF LINKS BETWEEN WASHINGTON D.C. AND OMAHA

To increase the versatility of these programs there must be provision for various data display options. The cathode ray tube is suitable for displaying items of transitory interest such as the man-computer dialogue, and to display graphical output such as network segments. A hard-copy facility must also be provided to permanently record the generated graphs, if desired, and to record voluminous tabular displays. Careful thought must be invested in designing these displays to insure that meaningful data is highlighted, and that less meaningful data is suppressed unless specifically requested.

The ground work for an effective and versatile data management and retrieval system for Strategic C^2 has been outlined above. Implementation issues revolving mainly around questions of host hardware can not be addressed. It is axiomatic that the larger and faster the host computer, the greater is the generality, flexibility and responsiveness that can be achieved. In order of increasing preference, the host computer could provide tape, disc, or drum storage, with combinations of these three preferable to any one singly. For interactive retrieval, various CRT or typewriter terminals are desirable, and a line printer for hard copy output is mandatory. The core requirement is a strong function of the desired amount of flexibility in the retrieval language as well as the ultimate length of the fixed records. If several users at remote sites are to have independent access to the data base, there must be terminals at these sites as well. Once the hardware availability has been determined, the final design details can be drawn -- then it is a matter of collecting the data and recording it on the storage medium. To be useful as a working tool, the data management and retrieval system must then be maintained in a constant state of timeliness, and there must be a continual program of refinement and updating. This means that notice of changes to the system be automatically sent from the originators of such changes to the data base librarian.

Conclusions

The steps towards implementing an effective Strategic C^2 data management and retrieval system have been outlined. The exact nature of the system will be dependent upon the hardware to be employed. However, even a rudimentary system could lead to improved understanding of the C^2 system. If used to advantage, the data base will not be a mere catalog of recorded information, but can present the user with an analytical tool of Strategic C^2 status and history. This tool can also be used by researchers at locations remote from the host computer. Extending this idea further, it could also be used as a standardized base of readily available inputs for all Strategic C^2 modelling and simulation efforts.

REFERENCES

1. Survivability Analysis of Command and Control Systems, Three Months Progress Report, AFOSR Contract No. F44620-72-C0080, Network Analysis Corporation, Glen Cove, New York. (August, 1972)

2. Ford, L. R. and D. K. Fulkerson, Flows in Networks, Princeton University Press, Princeton, New Jersey. (1962) pp. 130-134

3. Kendall, M. G. and A. Stuart, The Advanced Theory of Statistics, Vol. 2, Hafner Publishing Company, New York. (1967) pp. 98-160.

4. Kahn, H. and I. Mann, "Monte Carlo," Rand Report No. P-1165, Rand Corp., Santa Monica. (July 30, 1957)

5. Bennett, A. M., "The Use of Simulation Techniques in the Analysis of a Command, Control, and Communications System," Unnumbered MITRE Report. (1965)

6. "Calculation of SSPK Against a Target Via the Circular Coverage Function and the Physical Vulnerability System," report of AF/SASC.